

Application Platforms, Routing Algorithms and Mobility Behavior in Mobile Disruption-Tolerant Networks

Arezu M. Moghadam

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2011

©2011

Arezu M. Moghadam

All Rights Reserved

ABSTRACT

Application Platforms, Routing Algorithms and Mobility Behavior in Mobile Disruption-Tolerant Networks

Arezu M. Moghadam

Mobile disruption-tolerant networks (DTNs), experience frequent and long duration partitions due to the low density of mobile nodes. In these networks, traditional networking models relying on end-to-end communication cease to work. The topological characteristics of mobile DTNs impose unique challenges for the design and validation of routing protocols and applications. We investigate challenges of mobile DTNs from three different viewpoints: the application layer, a routing perspective, and by studying mobility patterns. In the application layer, we have built 7DS (7th Degree of Separation) as a modular platform to develop mobile disruption-tolerant applications. 7DS offers a class of disruption-tolerant applications to exchange data with other mobile users in the mobile DTN or with the global Internet.

In the routing layer, we have designed and implemented PEEP as an interest-aware and energy efficient routing protocol which automatically extracts individual interests of mobile users and estimates the global popularity of data items throughout the network. PEEP considers mobile users' interests and global popularity of data items in its routing decisions to route data toward the community of mobile users who are interested in that data content.

Mobility of mobile users impacts the conditions in which routing protocols for mobile DTNs must operate and types of applications that could be provided for mobile networks in general. The current synthetic mobility models do not reflect real-world mobile users' behavior. Trace-based mobility models, also, are based on traces that either represent a specific population of mobile users or do not have enough granularities in representing mobility of mobile users for example cell tower traces. We use Sense Networks' GPS traces

that are being collected by monitoring a broad spectrum of mobile users. Using these traces, we employ a Markovian approach to extract inherent patterns in human mobility. We design and implement a new routing algorithm for mobile DTNs based on our Markovian analysis of the human mobility. We explore how the knowledge of the mobility improves the performance of our Markov based routing algorithm. We show that that our Markov based routing algorithm increases the rate of data delivery to popular destinations with consuming less energy than legacy algorithms.

Table of Contents

1	Introduction	1
1.1	New applications and their requirements in mobile DTNs	7
1.2	Routing in mobile DTNs	8
1.3	Behavior and mobility patterns of mobile users	12
I	Application Platform for Mobile DTNs	16
2	New applications and their requirements in mobile disruption-tolerant networks	17
2.1	A brief introduction to the 7DS platform	18
3	A Modular Application Platform for Mobile DTNs	21
3.1	Disruption-tolerant applications provided by the 7DS platform	22
3.1.1	Web and email applications	22
3.1.2	Data sharing applications	23
3.2	Architecture and characteristics of the 7DS platform	27
3.2.1	Discovery module	28
3.2.2	Proxy server	29
3.2.3	Local web server	32
3.2.4	Search engine and cache manager	33
3.2.5	Multicast query engine	33
3.2.6	Mail transport agent	34
3.2.7	Data sharing module	34

3.3	Scalability of the 7DS platform	42
3.4	Related work	42
3.5	Conclusion	45
II	Content Distribution in Mobile DTNs	47
4	Interest-Aware Content Distribution Protocol for Mobile DTNs	48
4.1	The necessity of an interest-aware distribution mechanism	51
4.2	Related work	52
4.3	Interest-aware routing algorithm	56
4.3.1	Vector space model	58
4.3.2	Singular value decomposition	58
4.3.3	Interest-aware content distribution protocol	60
4.4	Simulation setup	60
4.4.1	Implementation of the interest-aware protocol in the ONE simulator	61
4.4.2	Reality-mining data traces	62
4.5	Evaluation of the interest-aware content distribution protocol	64
4.5.1	Evaluation of the protocol using participant-participant traces . . .	66
4.5.2	Evaluation of the protocol using Bluetooth-Bluetooth traces	69
4.6	Conclusion	69
5	PEEP: Popularity Based and Energy Efficient Content Distribution Protocol for Mobile DTNs	72
5.1	Feasibility of an interest-aware architecture for mobile DTNs	75
5.1.1	Interest-aware music and news sharing application	76
5.1.2	Popularity	77
5.2	PEEP distribution algorithm	78
5.2.1	Interest vectors	79
5.2.2	Transmit budget	79
5.2.3	PEEP algorithm	79
5.3	Simulation results	84

5.3.1	PEEP's implementation in the ONE simulator	84
5.3.2	Evaluation of the Results	89
5.4	Conclusion	91
III	Mobility Behavior of Mobile Users	93
6	A Markov Routing Algorithm for Mobile DTNs based on Spatio-Temporal Modeling of Human Movement Data	94
6.1	Introduction	95
6.2	Related work	97
6.2.1	Synthetic mobility models	97
6.2.2	From real-world traces to mobility models	98
6.3	Characteristics of our mobility traces	100
6.3.1	Extracting different features of human movement	101
6.4	Modeling mobile users' movements by N-grams	104
6.4.1	Bigram statistics	106
6.4.2	Trigram statistics	108
6.4.3	Evaluation of the N-gram models	109
6.5	Specifying a Markov chain for human movement	113
6.5.1	Characteristics of the Markov chain	114
6.6	A Markov based routing algorithm for mobile DTNs	118
6.6.1	Routing objective	118
6.6.2	Algorithm design	119
6.6.3	Evaluation of the algorithm by Monte Carlo simulations	121
6.7	Conclusion	126
IV	Conclusions	129
7	Conclusions	130

V	Appendices	136
A	Calculating Absorption Probabilities and Absorption Times for Markov Chains	137
A.0.1	Example	138
A.0.2	Example (Birth-and-death chain)	140
VI	Bibliography	142
	Bibliography	143

List of Figures

1.1	Mobile users' movement is a defining factor in the distribution of messages from one node to another until they reach from source S to the destination D. In this figure two communities of mobile users have been identified with different colors. 1 - source of the data, S, meets another mobile node and transfers a copy of the data to this node. 2 and 3 - More copies of the data is transferred to other mobile nodes as they meet. 4 - Finally, one of the mobile nodes which has a copy of the data meets the destination node, D.	4
1.2	Communication in mobile DTNs spans four major categories based on the direction of the information flow. The dark green blocks are implemented by the 7DS platform.	6
3.1	After publishing announcements at step 1 using BBS, mobile users can review and download the related information at step 2.	26
3.2	Modular architecture of the of the 7DS platform.	27
3.3	The algorithm of the 7DS proxy server for handling HTTP requests	30
3.4	The 7DS search page shows results for a keyword search. The results correspond to matching files in the local cache.	31
3.5	Mobile nodes discovering other peers and file information lists offered by them. Each file information embodies a hash value and date tuple.	36
3.6	Mobile nodes' shared folders after file synchronization.	37
3.7	File synchronization algorithm	39
3.8	Client generates signature file and server creates delta file according to the Rsync delta encoding algorithm.	41

4.1	Example of the interest-aware communication model. 1 - The source (S) estimates the potential recipients of data content (D) as members of the community Y . 2 - After calculating the similarity between the interests of the encountered users and the data content D , the data is transferred to the most similar user to group Y which is user b . 3 - Node b meets new mobile users. 4 - After similarity calculations, two mobile nodes e and g are selected as the recipients interested in the data content D	53
4.2	Communities of mobile users. Mobile users are assigned to appropriate communities of interest based on their interest-vectors (IV). Here, three communities with their respective interest-vectors, IV1, IV2, and IV3, in a multidimensional space, have been displayed.	57
4.3	Interest-aware content exchange 1 - Nodes i, j discover each other. 2 - Node i receives node j 's interest vectors. 3 - Node i relays document D to the node j , if node j 's interests match with the document D	59
4.4	Bluetooth device-to-device encounter tables as part of the reality-mining relational database. The MySQL table, represented as <i>person</i> , records the information about the owner of the device. The <i>devicespan</i> table contains encounters with other bluetooth devices either part of the reality-mining project or devices which belong to mobile users outside this experiment. . .	63
4.5	Coverage of the community of interest.	65
4.6	Distribution of irrelevant documents among mobile users.	66
4.7	Total number of dropped documents in mobile devices due to cache overflow.	67
4.8	Total aborted transfers due to insufficient contact times.	68
5.1	Example of different communities of interests. Our interest-aware music sharing application automatically shares music among users with similar taste in music.	75

5.2	Budget allocation methods in the PEEP algorithm. a) Interest-Only in which only items of interest are transmitted over the contact. b) Interest-Rand in which transmit budget is divided between items of interest and other randomly selected data items. c) Interest-EstPop and Interest-GlobPop in which the second portion of the transmit budget is assigned to transmit popular items.	78
5.3	Greedy behavior of the Interest-Only model. 1. The source (S) estimates the potential recipients of data content (D) as belonging to community Y . 2. After calculating the similarity between the interests of the encountered users and data content D , Interest-only model transfers the data to the most similar user to group Y , which is node b . 3. Node b meets new mobile users. 4. After similarity calculations, two mobile nodes are selected as the interested recipients of the data content, nodes e and g	80
5.4	Example of interest history window. Current index points to the most recent entry in the interest history buffer. Here, popularity is calculated based on the popularity estimation formula as the average of the interest vector entries in the buffer with $\alpha_i = 1$	83
5.5	Total number of distributed items of interest over time with medium to high cache size.	86
5.6	Normalized view of the rate of data distribution using different versions of PEEP with moderate cache size (500MB).	87
5.7	Normalized view of rate of data distribution using different versions of PEEP with small cache size (50MB).	88
5.8	Normalized rates of distribution of irrelevant items scaled in terms of encounters.	90
6.1	Map grid of San Fransisco with number of GPS pings at each grid.	102
6.2	A conceptual illustration of fitting Gaussian distributions on a user's movement data. In this example the user has 3 predominant locations where she spends 50%, 10%, and 25% of her time.	103

6.3	A snippet of a bigram table that is calculated from a user's movement data during a weekend. Bigram tables are calculated from the 80% portion of the movement data which is considered as train data.	106
6.4	A snippet of a trigram table that is calculated from a user's movement data during a weekend. Trigram tables are calculated from the 80% portion of this movement data that is considered as train data.	109
6.5	Log likelihoods which are calculated for each user's test data based on their workday n-gram tables.	110
6.6	Comparison between different n-gram models based on their average errors in calculating log likelihoods of users' test data.	111
6.7	Log likelihoods which are calculated for aggregated users' data from Figure 6.5. In this figure users are classified together based on their predominant locations.	112
6.8	An example of a user's Markov chain which is extracted from a user's workday movement trace. This Markov chain is also ergodic as higher powers of its transition matrix converges to a stationary distribution.	114
6.9	An example of a user's Markov chain which is extracted from a user's workday movement trace. This Markov chain turns out to be nonergodic as higher powers of its transition matrix does not converge to a stationary distribution.	117
6.10	Transmission delay vs. consumed energy for a randomly selected destination.	124
6.11	Transmission delay vs. consumed energy for a popular destination or hotspot.	125
A.1	Markov chain of example A.0.1.	138
A.2	Birth-and-death-chain	140

List of Tables

6.1	Comparison of the movement traces that have been used in trace-driven studies	128
-----	---	-----

Acknowledgments

I am grateful to my thesis advisor, Prof. Henning Schulzrinne, for his guidance, support, valuable insights, and discussions in the course of my study at Columbia. His enthusiasm and faith in this research combined with his insistence in high-quality scientific venture were a strong motivation and driving force; it was an honor to work with Prof. Schulzrinne. I would also like to thank my dissertation committee members, Prof. Henning Schulzrinne, Prof. Tony Jebara, Prof. Vishal Misra, Prof. Dan Rubenstein, and Dr. Venkatesh Krishnaswamy, for their valuable feedback and examination of my dissertation.

Studying at Columbia has been an exceptional education experience and I am grateful to all the professors who taught me the principles and the state of the art in my discipline. It was a wonderful opportunity to know and work with faculty members of the Department of Computer Science and current and former members of the Internet Real Time Lab (IRT): Knarig Arabshian, Salman Abdul Baset, Sangho Shin, Kundan Singh, Vishal Singh, Xiaotao Wu, Kenta Yasukawa, Weibin Zhao, Omer Boyaci, Ashutosh Dutta, Se Gi Hong, Jong Yul Kim, Jae Woo Lee, Kumiko Ono, Charles Shen, Wonsang Song, and Suman Srinivasan; I thank all of them for making IRT such a great place to work. It was a great joy to share my office, Schapiro 720, with Wisam Dakka, Sameer Masckey, Kumiko Ono, and Yves Petinot which ultimately became the beginning of our friendship. Thanks are due to the staff of the Department of Computer Science for their help and support during the course of my study.

I would also like to thank Dr. Janet L. Kayfetz for teaching me how to be a better writer and a more confident presenter, for her patience in reading and correcting my first paper, and for being a great support during my PhD. I also would like to thank all of my friends which made my life in New York such a pleasant experience: Saba, Gabriela, Astrid, Narges, Nasim, Shaadi, Shabnam, Nicola, Matei, Sameer, and Hessam; I thank them all for

being there for me in joy and hardship.

I would not have been able to achieve this much without the support and encouragement of my family and I am delighted to dedicate this thesis to all of them. I am thankful to my late mother, Parivash, for teaching me the principles of love and excellence. I am indebted to my father Rauf, for his love, support, and sacrifices over the years. My parents always stood by me and their vision and confidence in me have been true inspiration. I am also grateful to my sister, Elham, for her love, support, encouragement, and being an understanding listener at times of desperation. I also would like to thank my aunt, Pari, for her love and support throughout these years.

New York, NY

May 2011

To My Parents

Chapter 1

Introduction

The applications and supporting protocols of today's Internet depend on a pre-established end-to-end communication path to route traffic between two parties involved in a communication. These applications and protocols function poorly when faced with operating environments that suffer from frequent network partitions or unstable paths with very long delays. Unstable paths can be the result of several challenges at the link layer, for example, high node mobility, low node density, and short radio range (intermittent power from energy management schemes, environmental interference and obstruction and denial-of-service attacks [Fall, 2003], [Burgess *et al.*, 2006]). We classify these types of networks as disconnected or disrupted vs. the connection-based model which is built on the assumption that the source and destination are connected for the duration of the communication session [Cerf *et al.*, 2007], [Fall, 2003]. Such environments can exist in undeveloped areas or when a stable infrastructure is destroyed by a natural disaster or military actions. These networks are assumed to experience frequent, long-duration partitioning and may never have an end-to-end instantaneous path between communicating hosts. Disruption-tolerant networks (DTNs) are useful when the information being routed retains its value longer than the disrupted connectivity which delays the delivery. In this dissertation, we specifically focus on a special type of these networks where network partitions are due to different forms of host mobility. In these networks, end-to-end communication paths between mobile hosts can not be sustained when hosts are away from a global network infrastructure. Such networks are categorized as mobile DTNs.

We study the specific characteristics of disconnected mobile networks and define and implement a set of mobile disruption-tolerant applications which assist mobile users to exchange information in a local setting or with the global Internet. Having an efficient routing algorithm is a key factor in enhancing users' experience in better utilizing the applications in all networking scenarios. Therefore, after discussing mobile DTN applications we turn our focus to design and implementation of more efficient routing algorithms for mobile DTNs. Since users' mobility is the main drive for communication among users in mobile DTNs, having better knowledge about human mobility leads to better design of the routing algorithms. Therefore, studying different aspects of human mobility is the last focus of discussion in this thesis.

With the growing number of mobile devices equipped with different wireless technologies, such as IEEE 802.11, Bluetooth, and other radio interfaces, situations where communication is desirable can occur at any time and any place, even where no networking infrastructure is available. As a result of these advances in various communication means, mobile users increasingly find themselves in different types of networking environments. These environments, ranging from globally connected networks such as cellular networks or the Internet to the entirely disconnected networks of stand-alone mobile devices, impose different forms of connectivity. Due to users' mobility and migration from globally connected environments to isolated stand-alone mobile networks, mobile users experience disruption in their communication services. However, as global communication services are disrupted, new local communication opportunities with other mobile users or stationary, stand-alone data repositories might become available. While mobile users move, they may find themselves in the wireless range of other mobile users and, although away from a networking infrastructure, their devices can communicate in a peer-to-peer fashion. These locally-formed mobile networks, which can last for seconds to hours, provide users with new communication opportunities to exchange different types of information such as web pages, music, video, news, email, and emergency alerts.

Mobile DTN technologies might be useful even in the presence of a global cellular data network. Even when data links to a cellular network are available, mobile users can still take advantage of the local WiFi contacts to exchange large amounts of data faster and in

a more energy efficient fashion than transmitting data over GSM. In fact, power analysis of different wireless interfaces shows that for high transmission rates GSM radio consumes considerably more power than WiFi radio [Balani, 2007].

Traditional, "connected" applications rely on interactive protocols in which a complete one-way application message might contain many source-to-destination signaling roundtrips at the transport layer. However, the types of the applications that are operable in mobile DTN environments are different from the traditional applications. The legacy connected applications cease to work in disruption-tolerant environments due to the abrupt disconnections typical of such networks. The absence of the end-to-end connectivity in these types of networks requires a different set of transport and application layer functionalities. Therefore, a new class of mobile applications has to be defined and a new set of routing algorithms has to be devised to be useful in mobile DTNs. In order to develop applications and routing algorithms which could recover from disruptions, specific characteristics and core requirements of communication in mobile DTNs needs to be identified.

Routing data from a source to its destinations is a fundamental requirement for all communication networks. In mobile DTN scenarios, routing becomes a challenging task due to the lack of infrastructure and frequently changing topology. Even, popular ad-hoc routing protocols such as AODV [Perkins *et al.*, 2003] and DSR [Johnson *et al.*, 2007] fail to establish on-demand routes. This happens because these protocols try first to establish a complete route and then, after the route has been established, forward the actual data. Therefore, in Mobile Adhoc Networks (MANETs) communication is still classified as being end-to-end. However, due to characteristics of mobile DTNs, instantaneous end-to-end paths are difficult or impossible to establish. Instead of relying on end-to-end network connectivity, DTNs take advantage of temporary connections to relay data in a fashion similar to the postal network [Fall, 2003]. Routing in DTNs must take a "store-carry-forward" approach, where data is incrementally moved and stored throughout the network, hoping it will eventually reach its destination [Fall, 2003], [Burgess *et al.*, 2006]. Therefore, routing is performed incrementally and over time with no guaranteed delivery.

Store-carry-forward communication exploits nodes' mobility to bring messages closer to their destinations by exchanging them with other mobile nodes when they meet, as

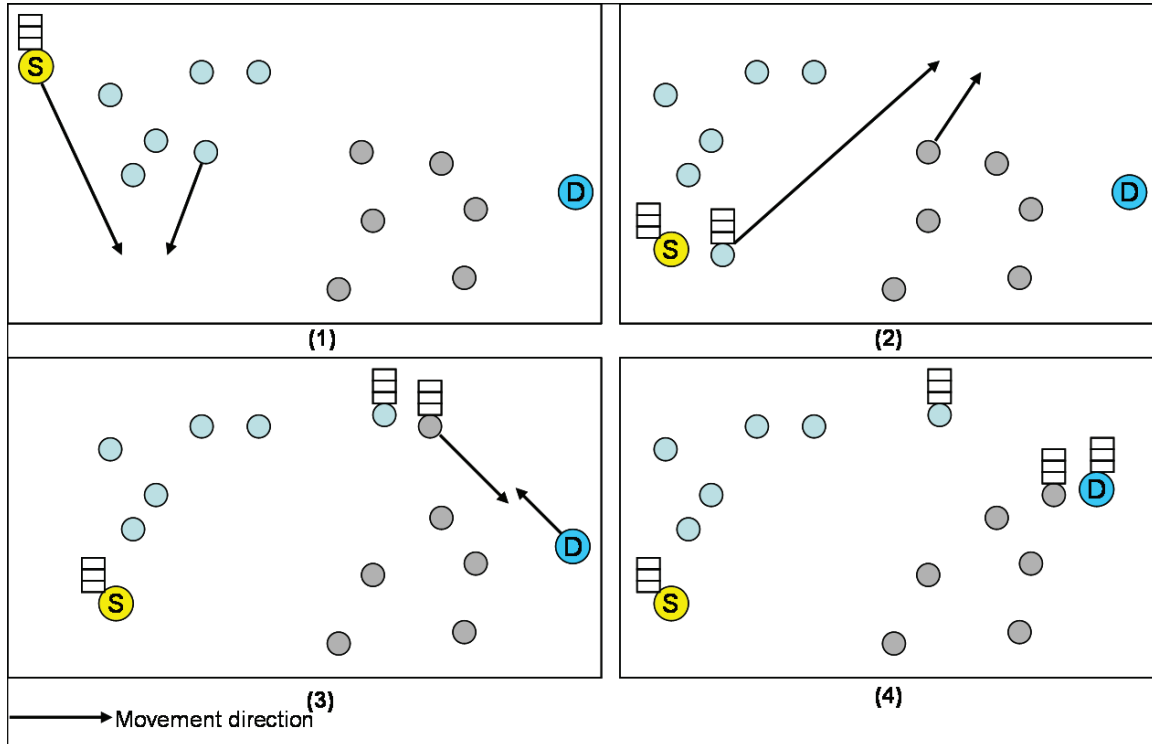


Figure 1.1: Mobile users' movement is a defining factor in the distribution of messages from one node to another until they reach from source S to the destination D. In this figure two communities of mobile users have been identified with different colors. 1 - source of the data, S, meets another mobile node and transfers a copy of the data to this node. 2 and 3 - More copies of the data is transferred to other mobile nodes as they meet. 4 - Finally, one of the mobile nodes which has a copy of the data meets the destination node, D.

illustrated in Figure 1.1. Therefore, node mobility is a defining factor in routing of the messages to their final destinations. Mobility impacts the conditions in which routing protocols must operate and types of applications that can be provided for mobile DTNs. As a result, performance of the routing protocols and mobile applications highly depends on the mobility behavior of mobile users who operate in the disconnected mobile network. Another major application of modeling the mobility behavior is simulations. Simulations are the main tools to evaluate the performance of new routing protocols and applications. Most researchers use simulations to evaluate how their applications, systems or protocols respond to variations in user activity, most importantly mobility. Therefore, meaningful evaluation of different routing algorithms for mobile DTNs heavily depends on the choice of the mobility model [Chaintreau *et al.*, 2007]. Moreover, types of the applications and routing that could be provided for mobile networks depend on the type of mobility of mobile users. Mobility defines the rates in which network topology changes and how often mobile nodes encounter or stay contacted. The duration of contacts and the type and amount of data that can be exchanged by mobile users, therefore, depend highly on the mobility behavior of mobile users. In other words, mobility defines the characteristics of the inter-contact times among mobile users. Consequently, the types of applications and services are also defined by the mobility of mobile users. For instance, the applications and routing technologies for vehicular networking are intrinsically different from applications and routing mechanisms for a network of pedestrians.

As the above discussion suggests, applications, routing algorithms, and mobility models introduce three challenging areas in mobile DTNs. The special characteristics of mobile DTNs and their transient topology have left many unresolved problems in all these three areas that must be resolved before a successful realization of mobile DTNs can be achieved. Therefore, this thesis discusses problems in mobile disruption-tolerant networks from three different viewpoints: applications, routing, and mobility patterns of mobile users.

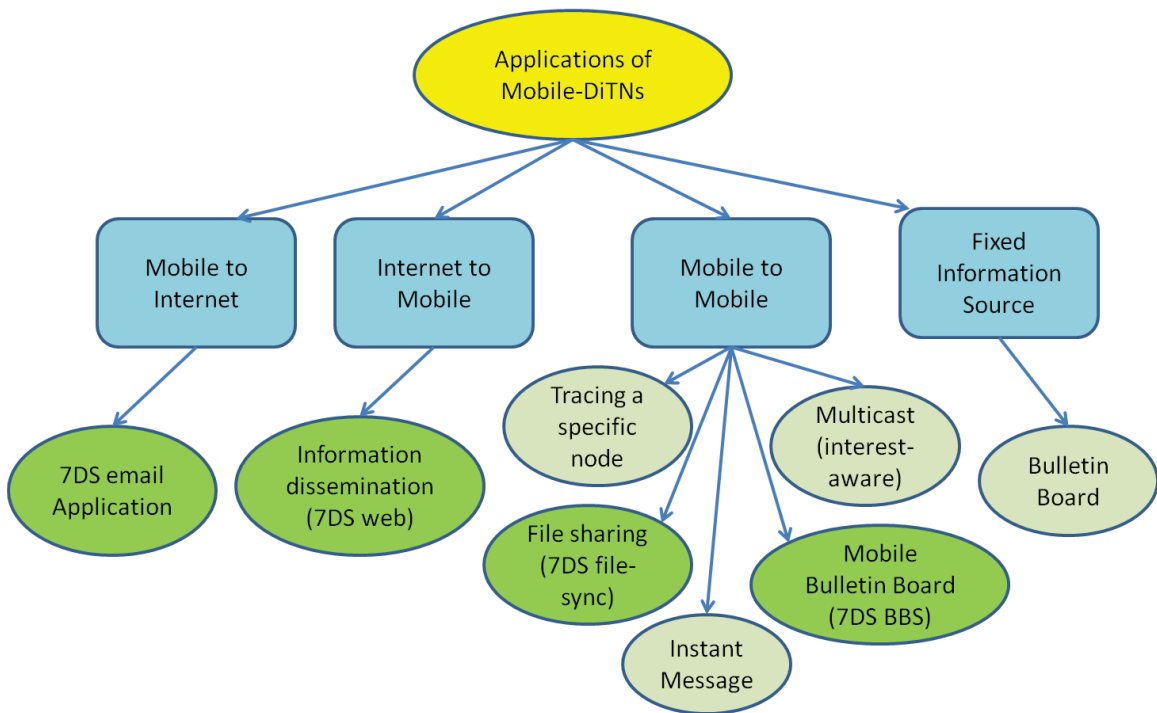


Figure 1.2: Communication in mobile DTNs spans four major categories based on the direction of the information flow. The dark green blocks are implemented by the 7DS platform.

1.1 New applications and their requirements in mobile DTNs

Existing "connected" applications which rely on interactive transport protocols are not directly reusable in mobile DTNs. Furthermore, the concept of always-on servers is also unreasonable in these environments, as the topology of the network is frequently changing because of the mobility of mobile users. Therefore, in these scenarios, mobile users have to rely on node, and service discovery, and applications that cope with network disruptions. In the 7DS project (7th Degree of Separation) [Srinivasan *et al.*, 2007], [Moghadam *et al.*, 2008], we have been investigating ways to extend core Internet services such as web and email to new web applications for mobile DTNs. The first requirement for the Internet applications to be extendable to these types of mobile networks is that they should be disruption-tolerant, meaning their services must not depend on an always-on Internet connection.

Due to the special characteristics of DTNs, they require a different set of functionalities in the transport and application layers. 7DS was developed to provide these necessary functionalities for mobile nodes to exchange information in a store-carry-forward fashion. The 7DS platform provides an abstract layer to conceal networking details and disruptions from the applications. We have defined a class of disruption-tolerant applications based on the communication model they employ and the services they provide, as illustrated in Figure 1.2. As shown in Figure 1.2, communication in mobile DTNs spans four major categories based on the information flow from mobile nodes to the Internet, Internet to the mobile nodes, peer-to-peer communication among mobile nodes, and communication between stationary information centers such as bulletin boards and mobile nodes.

The choice of applications is important in identifying the required core components of the 7DS platform. 7DS applications cover a broad range of core requirements in device and resource discovery, zero configuration network setup, store-carry-forward routing, data searching, messaging, and data sharing. The 7DS platform, as an application platform for mobile DTNs, is able to provide disruption-tolerant versions of two core Internet services, namely web access and email. 7DS's web application enables mobile users to retrieve web objects from the Internet, even if they are disconnected from a global network. Mobile users are also able to deliver email messages to the Internet by using 7DS's disruption-tolerant email application. In addition, 7DS offers peer-to-peer communication among

mobile users by its file synchronization applications and communication with stationary and mobile data centers through its bulletin board system (BBS). Furthermore, we have extended 7DS [Moghadam *et al.*, 2008] as a modular platform which provides core underlying functionalities required to develop new mobile disruption-tolerant applications. Application developers can use our 7DS's core APIs created as part of this research to develop new applications by using different 7DS modules and without worrying about the underlying network setup and communication.

Chapter 3 explains the 7DS architecture in detail. We further discuss a class of disruption-tolerant applications that includes web query, email, file synchronization and bulletin board system that we have implemented for 7DS. By considering these applications, we have determined a common set of primary functionalities that should be provided by 7DS as a modular application development platform. We explain how we have evolved 7DS toward a generic software platform that provides application developers with these underlying functions. In addition, we explain by examples how to use 7DS as a software platform to develop different applications for mobile DTNs.

1.2 Routing in mobile DTNs

The routing problem in a DTN may at first appear like the standard problem of dynamic routing in the Internet (RIP [Malkin, 1998], OSPF [Moy, 1998]) but with extended link failure times. For the standard dynamic routing problem, however, the topology is assumed to be connected (or partitioned for very short intervals), and the objective of the routing algorithm is to find the best currently-available path to move traffic end-to-end. In DTNs an end-to-end path may be unavailable at all times. Routing, hence, is performed over time to achieve eventual delivery by employing long-term storage at the intermediate nodes. Therefore, the routing problem in DTNs becomes a shortest-path problem where connection links may be unavailable for extended periods of time and a storage constraint exists at each node. This formulation turns DTN routing into a difficult optimization problem [Jain *et al.*, 2004]. Different routing algorithms that are developed for mobile DTNs try to mitigate the absence of the end-to-end connection by exploiting mobility and opportunistic wireless

contacts to transfer data closer to its destination at each hop. Therefore, the routing problem in mobile DTNs is addressed as a per hop routing rather than end-to-end routing.

Most of the routing algorithms for mobile DTNs solve the data routing problem between a source and a specific destination [Lindgren *et al.*, 2003], [Zhao *et al.*, 2004], [Thakore and Biswas, 2005], [Jones *et al.*, 2005], [Boldrini *et al.*, 2007], [Burgess *et al.*, 2006]. These routing algorithms that follow a one-to-one paradigm for routing use some heuristics or probability measures to estimate the best possible hop-by-hop route from source to the destination node. However, in many real-world scenarios it is necessary to extend data transmission from individual recipients to a group of recipients. Scenarios such as propagating news or advertisements, managing emergencies, and sharing images or documents among members of a community are examples of communication with a group of recipients. This routing problem for communication with groups of mobile users is similar to the problem of multicasting in the Internet. Proposed models for multicasting data through mobile DTNs, in fact, try to extend the classical multicast model of the Internet to the mobile disruption-tolerant networks. They define new multicast semantics to overcome the intrinsic network partitioning in these networks [Zhao *et al.*, 2005], [Chen *et al.*, 2006]. However, these proposed architectures require a global knowledge of the group memberships of mobile users and the network topology [Zhao *et al.*, 2005]. As the result, implementing these architectures is not feasible in mobile DTNs which suffer from a frequently changing topology and lack of infrastructure to track group memberships.

So far, the practical architecture for multicasting in mobile DTNs has been epidemic multicast [Vahdat and Becker, 2000] in which all mobile users flood the other mobile users with all the data in a hope and this data eventually finds a route to the intended recipients. The main problem with epidemic routing is the inefficient usage of communication resources such as storage space, transmission energy, and bandwidth. Although storage space usage can be overlooked, considering the dramatic increase in mobile storage space, energy and bandwidth are still important restrictions. Today, mobile devices are equipped with a variety of functionalities such as wireless interfaces, GPS, and multimedia. All these peripheral equipments consume energy and place growing pressure on the device's battery life. On the other hand, advances in battery design have stalled for the past few years [Balani, 2007].

With recent advances in mobile multimedia communication, efficient usage of bandwidth is considered as an important design criterion in addition to energy in developing new mobile technologies.

Routing algorithms that are designed for mobile DTNs transmit data to its destination through intermediate nodes, also known as relays, which might not be interested in the data themselves. Therefore, mobile relays might use their communication resources to carry data without any personal gain. The reliance of the mobile DTN routing protocols on mobile users' communication resources to carry other users' data with no specific personal benefit is an important burden for the wide deployment of mobile DTN technologies. Mobile DTN technologies could benefit from more intelligent routing algorithms which could help the intermediate mobile users locate the data items they are interested in while routing other users' data.

Selecting the relays based on some notion of community is one design solution that could help all intermediate parties involved in a communication benefit from the routing. In this concept, community is defined based on some similarity measure between mobile users. This, of course, assumes that mobile users in the same community are interested in similar content. For example, tourists who are visiting a city might be interested in similar sightseeing information. Therefore, if the mobile users act as the relays of the data for their own community, they also benefit from carrying that data themselves.

Community could have a broader definition derived from different behaviors of mobile users. For example, Hsu et al. [Hsu *et al.*, 2008] have proposed an architecture that selects the relays by relying on similarity of the mobility profiles between mobile users. Mobility profiles are defined based on mobile users' historical patterns of visiting different geographical locations. Hsu et al.'s routing algorithm distributes a message among groups of mobile users with similar mobility patterns. This model, however, is oblivious to the fact that users with correlated mobility patterns might not share any common interests. Therefore, the data received by relays might not be useful for them. For instance, different groups of tourists visiting a city might not be interested in the same restaurant or movie reviews, even though they had visited similar locations in the past.

A more intelligent content distribution algorithm for mobile DTNs should be able to

learn mobile users' interests over time. After learning mobile users' interests, this algorithm should be able to categorize mobile users based on their interests and distribute data among corresponding interest groups. This *interest-aware* algorithm is basically a multicast algorithm for mobile DTNs that does not require any previous knowledge of group memberships or network topology to perform correctly. In Chapter 4, we propose such an interest-aware algorithm that automatically learns users' behaviors, extracts their preferences, and classifies them into appropriate interest categories. Our algorithm learns users' interests from their cached data and represents these interests in terms of *interest-vectors* and uses these vectors to route data to recipients who are interested in that data. When two mobile users meet, they exchange their interest-vectors. They use these vectors to calculate each other's interest in the content they are trying to distribute. The data is then transferred to the other user only if there is enough correlation between the user's interests and the content of the data. Therefore, unlike epidemic multicast, data is propagated throughout the network only through the intermediate users who are also interested in the data content. In Chapter 4, we will show that our interest-aware routing algorithm produces a significant improvement over epidemic routing in locating and disseminating data among communities of interest.

In Chapter 5 we discuss the inadequacies of the interest-aware algorithms. The main problem with the interest-aware algorithms is that they act greedy—meaning that they just consider the interests of the mobile users in the immediate first-hop wireless contacts. This prevents mobile users who are also interested in the data from receiving it because they are a few hops further down the sequence of transmission. Therefore, interest-aware routing mechanisms may cause the entire network to starve when all nodes act greedily. Another issue with the interest-aware algorithms is its inefficient usage of energy and bandwidth because it does not limit data transmission although energy and bandwidth resources are limited.

In Chapter 5 we evolve the interest-aware communication model toward a more energy and bandwidth efficient model which we have named PEEP. PEEP solves the interest-aware greediness and energy consumption issues. We solve the energy issue by introducing transmit-budgets. The transmit-budget is a data transmission token which determines the

amount of data allowed to be transferred over each wireless contact. The transmit budget is determined based on the policies enforced by the user or application. We also propose different transmit-budget allocation methods in the PEEP algorithm to avoid the greedy behavior of the interest-aware communication model. The main idea behind these allocation methods is to divide the transmit-budget into two portions. The first portion is dedicated to transfer items of interest for the mobile users who are involved in the current wireless contact. The second portion is allocated to transfer other data items which are estimated to be useful for the mobile users who are encountered in the future. By transferring *other* data as well as data of interest at each contact opportunity, PEEP solves the greedy behavior of the interest-aware algorithms. We will implement both interest-aware and PEEP algorithms and compare their performance with the epidemic multicast in Part II.

1.3 Behavior and mobility patterns of mobile users

In mobile DTNs, it is the mobility of nodes that provides new paths of communication. Unlike traditional, connected applications that suffer from network disruptions caused by node mobility, the emerging class of mobile disruption-tolerant applications may turn mobility into an advantage. For example, in traffic control applications developed for vehicular networks, the high speed of mobile nodes (vehicles) can help to inform drivers who are miles away about congestion or other emergency situations fast. In other scenarios, content-sharing applications can exploit opportunistic wireless contacts between passengers who are riding on a train, to exchange different information such as news, video or audio files. When these passengers get off the train and walk to other neighborhoods, these files could be distributed among other pedestrians present in those locations. By considering these scenarios, we recognize that different forms of mobility impose different forms of networking conditions. Moreover, the types of applications and routing protocols and their requirements differ depending on the type of the mobility of mobile users. Therefore, the knowledge of the mobility of mobile users assists mobile service and content providers in delivering better content and better application provisioning.

Understanding mobility models of mobile users is especially useful for simulation pur-

poses. Simulations are the main validation tool of new applications and routing protocols. The value and correctness of this validation, however, depends highly on how realistic the mobility models used in the simulations are. Therefore, having a realistic understanding of movement patterns and human interactions is a crucial factor in the practical design and validation of the routing protocols for mobile DTNs. There are two major classes of mobility models that are used in the simulations. These models are either produced synthetically, or they are based on real-world traces. Trace-based mobility models are extracted from movement traces of real-world mobile users. However, a very limited number of available real-world traces are used in the mobile networking research and simulations, and these traces have mostly been collected from movement scenarios of certain types of mobile users such as traces of students visiting different WiFi access points on campus. Because of the limitations of the trace-based mobility models, synthetic models are used for most simulation purposes. These synthetic mobility models are popular. They are easy to use to produce different simulation scenarios with arbitrary numbers of mobile users by changing only the parameters of the mobility model. However, most widely used synthetic models are very simplistic, because their main focus is their ease of implementation. In addition, these synthetic models are generated based on random parameters which might differ significantly from the reality. Consequently, simulation results of protocols based on these mobility patterns may differ significantly from real-world performances.

Traced-based mobility models also suffer from a lack of real-world large-scale mobility traces. The movement traces that have been used to extract these models are mostly collected by monitoring a specific subset of mobile users [Henderson *et al.*, 2004], [Mcnett and Voelker, 2005], [Mining, 2003]. Therefore the synthesized models generated from these traces are not easily extendable to general classes of users. The larger-scale traces, collected from cell towers, are either difficult to obtain because of the privacy issues or they don't have enough precision, as they cannot reflect the mobile nodes movements within the range of a tower which might extend up to 20 miles.

In Chapter 6 we explain the Sense Networks' mobility traces [SenseNetworks, 2008] that we have used in our mobility behavior studies. Sense Networks, Inc. indexes the real world using real-time location data for predictive analytics. Sense Networks' GPS traces are

being collected by monitoring a general population of mobile users from different cities in the United States. In Chapter 6, we describe our methods for extracting different statistical aspects of human movement from these traces. We also introduce new directions to derive the spatial and temporal patterns latent in human movement. The inherent pattern in human mobility is useful for different aspects of service provisioning in mobile networks ranging from the prevention of epidemics to urban planning [Kleinberg, 2007]. We explain how extracting patterns and regularities in human movement can lead to the design of new data distribution algorithms for mobile DTNs.

Furthermore, we extract statistical models to estimate each user’s most probable locations in the future. In our model, human movement is represented as a sequence of consecutively visited locations. Each element of this sequence is the location where the user was located at the time of the GPS ping. With the use of this representation, the problem of predicting a user’s future move converts to the problem of predicting the next item in the sequence. We use n -gram models [Brown *et al.*, 2001], which are popular in natural language processing (NLP) to calculate users’ future moves. N-grams that are used in various areas of natural language processing and in genetic sequence analysis are a type of probabilistic model for predicting the next item in sequences of words or DNA respectively. We have adapted n -grams for modeling the sequence of human moves.

Based on our statistical analysis of the empirical data, mobile users’ movement can be best described by using bigrams or second-order Markov models or Markov chains. Modeling human movement by Markov chains enables us to investigate different aspects of human movement using classical Markov methods. We investigate different features of these Markov chains such as ergodicity, and we also calculate absorption times and the absorption probabilities of the mobile users to different locations. Moreover, we use absorption time calculations to design and develop a new Markov-based routing algorithm for mobile DTNs. We implement and evaluate this new routing algorithm using simulations. We also implement a synthesized movement generator engine based on the Markov chains we extract from mobile users’ movements. Because of its probabilistic nature, this movement generator is built by generatively sampling from the movement data and is used to provide a simulation environment to validate new routing protocols and applications. We use this

movement generator engine to synthetically recreate the mobility of the mobile users based on Sense Networks' traces in our simulation scenarios. Our simulation results show that our Markov-based routing algorithm is most suitable for routing data to destination locations that are popular. Popular locations or hot-spots are neighborhoods that are more frequently visited by mobile users such as those that contain attractions and thus frequently visited by tourists.

Part I

Application Platform for Mobile DTNs

Chapter 2

New applications and their requirements in mobile disruption-tolerant networks

The problem of application design is especially challenging in mobile DTN environments as DTN applications can not depend on any infrastructure or always-on server. Moreover, the data that is exchanged by these applications should keep its value for more than the duration of network disconnection or delay in data delivery. In this chapter and the next one we study mobile DTNs from the application's viewpoint and we introduce the 7DS system and its modular design as a generic platform for application development in mobile DTNs. We explain 7DS's [Papadopouli and Schulzrinne, 2001a; Papadopouli and Schulzrinne, 2000; Papadopouli and Schulzrinne, 2001b] original concept and how we evolved it toward a new platform to develop mobile applications for mobile DTNs. 7DS was originally designed as a P2P data dissemination and pre-fetching tool for mobile users in the absence of a global Internet connection. We evolved 7DS toward a generic modular platform that provides application developers with core underlying functionalities required for communication in mobile disruption-tolerant environments. 7DS now provides the necessary transport and application layer functionalities for mobile nodes to exchange information using store-carry-forward communication paradigm.

The new 7DS system provides two core Internet service functionalities in the application layer, namely web access for information retrieval and email for delivering messages from mobile nodes to the Internet. 7DS enables web page and email exchange among mobile users by implementing a modular architecture that includes a proxy server, a multicast query system, a search engine, and an email transport module. In addition to the Internet services, 7DS makes file and event sharing in disconnected networks possible, by providing another set of services in the application layer that enable exchange of information between peer mobile devices. These applications assist mobile users both in file and event sharing in a server-less and disconnected mobile networking environment.

We have implemented a 7DS prototype system that leverages opportunistic contacts and P2P communication to build a scalable system that can deliver data to and from mobile nodes. 7DS can efficiently and transparently exchange data among peers in the absence of a global network connection. Data exchange with the larger Internet occurs when mobile devices can enter WiFi hotspots. Local data exchange is possible by data synchronization and bulletin board system applications that are developed using modules and protocols provided by the 7DS platform.

2.1 A brief introduction to the 7DS platform

In mobile ad-hoc and mesh networks mobile users depend on P2P wireless encounters for end-to-end communication and data exchange. Therefore, in these types of mobile networks when node density decreases below the necessary level to sustain ad-hoc and mesh networks, mobile devices experience disruption in their communication service. The two main reasons for disruption in communication are node mobility and the absence of any global wireless communication service such as an access point in WiFi or a base station in cellular networks. In mobile DTN scenarios, mobile devices may continue data exchange in a local wireless network setting by transitioning to message-based communications and leveraging node mobility. 7DS (Seven Degrees of Separation) [Srinivasan *et al.*, 2007], [Moghadam *et al.*, 2008] has been developed as a platform to address connectivity problem in these types of mobile networks by providing store-carry-forward communication [Fall, 2004]. We

originally developed 7DS [Papadopouli and Schulzrinne, 2001a], [Srinivasan *et al.*, 2007] as a bundle of applications for web query and email exchange in mobile disruption-tolerant scenarios. Implementing these applications revealed the need for a more standard software development environment. Such an environment should facilitate application development by providing the underlying functions required for communication in disconnected networks [Cerf *et al.*, 2007] such as device discovery and connection set-up. Therefore, we have extended 7DS as a modular application development platform that also provides mobile nodes with disruption-tolerant applications.

7DS allows for successful exchange of relevant information based on two realistic assumptions: 1) that devices move in and out of the locally constructed wireless network, eventually connecting to some global network (e.g, the Internet) and sending out information and bringing in new information and 2) that there is a high probability that the information that is needed by some user exists on a device in the near vicinity, for example, mobile users can share the most recent news articles with other mobile users who do not have access to a global Internet connection. (The second assumption is based on the fact that there are globally popular data items that would interest most users.)

The information flow in a network, in the absence of any infrastructure and server could be facilitated by a P2P data exchange system. Therefore, the 7DS system should be capable of setting up a P2P wireless communication network that uses very little bandwidth and is also very robust. It should also be able to work seamlessly in a highly mobile scenario where users are moving in and out of the local and stand-alone networks of mobile devices that are disconnected from the global Internet. It also has to be interoperable, platform-independent, and use resources sparingly to enable it to run on a variety of devices, from embedded systems to laptop computers.

As well as being a communication system, 7DS has been developed as a platform to provide necessary core components for disruption-tolerant application development. The Evolution of 7DS toward a generic software platform, was motivated by our earlier work on developing mobile applications for mobile disconnected networks [Srinivasan *et al.*, 2007]. This platform provides an abstract layer to conceal networking details and disruptions from the applications. The choice of applications is important in identifying the required core

components of the platform. In the next chapter, we enumerate a class of disruption-tolerant applications that includes web query, email, file synchronization, and a bulletin board system. By considering these applications, we determine a common set of primary functionalities to be provided by our software platform. These applications cover a broad range of core requirements in device and resource discovery, search engine, messaging, and data sharing. We explain how we are evolving 7DS toward a generic software platform that provides application developers with these underlying functionalities. In addition, we explain how to use our new software platform to develop new applications for mobile, disconnected networks.

Chapter 3

A Modular Application Platform for Mobile DTNs

In this chapter we discuss the fundamental requirements for mobile applications to be operative in disruption-tolerant environments. Then, we explain the architecture of the 7DS system and how it has been designed to satisfy these requirements. 7DS is a modular software development platform for mobile DTNs which also offers various disruption-tolerant applications. Each 7DS module implements a core functionality required for operation in mobile disruption-tolerant environments. 7DS provides an abstract API to the higher level modules and applications, in order to expand the 7DS software platform in a modular fashion. In Section 3.1 we address a class of disruption-tolerant applications and discuss their fundamental requirements to be disruption-tolerant, meaning that they are able to continue their service even in the absence of a global network connection. Section 3.2 elaborates on the 7DS system and its modular structure that allows it to construct these applications. We also explain how application developers can use the components and APIs of the 7DS platform to develop mobile disruption-tolerant applications. The modular design of the 7DS architecture facilitates future additions to this platform that are independent of the core system. Section 3.3 discusses the scalability considerations of the 7DS system. We review the related work in this area and their differences and similarities to the 7DS platform in Section 3.4. Finally, Section 3.5 concludes this chapter.

3.1 Disruption-tolerant applications provided by the 7DS platform

In the 7DS project we have been investigating how to extend core Internet services to mobile DTNs. The first requirement for the Internet applications to be extendible to these types of mobile networks is they should first of all be disruption-tolerant, meaning that their services must not depend on an always-on Internet connection. The 7DS platform offers applications that meet this requirement. 7Ds is able to emulate two core Internet services, namely, web access for information retrieval and email for delivering messages from mobile nodes to the Internet. Web access emulation is accomplished by implementing a special proxy agent while email access is achieved through implementing a SMTP server [Klensin, 2001] in the 7DS platform.

The ultimate goal of the mobile disruption-tolerant applications is to provide information accessibility for all mobile nodes operating in disruption-tolerant networks. Therefore, another important class of applications that should be devised for these environments is data sharing. In other words, mobile devices should be able to share files, news, video, audio or other useful information with each other in a local connection setting.

In this section, we introduce different applications that we have developed for the 7DS platform. We further explain how these applications cover a wide range of core functionalities. Later, in Section 3.2, we illustrate the components that have been added to the 7DS architecture to realize these applications. We have implemented these applications for Linux and Windows platforms and have released the software for public use [Moghadam *et al.*, 2006].

3.1.1 Web and email applications

7DS web and email applications are designed with consideration of two realistic assumptions: 1) Devices moving in the disconnected mobile network bringing in new information, and devices moving out of the network eventually connect to the Internet and send out information. (2) There is a high probability the information being searched for exists on a device in the near vicinity.

Based on these assumptions, in the absence of a connection to the Internet, mobile users' web queries-especially for popular items-can be fulfilled from the data that is available inside the local network on mobile users' web caches. The 7DS web proxy server redirects a mobile users' exact HTTP requests for different web-pages to other mobile users in the local network. If a requested web-page is not found within the local network, the 7DS proxy server presents mobile users with 7DS's query web-page. Mobile users can submit their search keywords through this web-page to other devices in their vicinity for different web objects in which they might be interested until some connection to the Internet becomes available (e.g., visiting an access point). The 7DS platform uses a very lightweight protocol involving simple XML messages for exchanging search keywords and their corresponding responses with peers. When disconnection from a global network occurs, the 7DS platform automatically multicasts the keyword queries to other mobile peers, retrieves the responses, and presents the user with the requested data. Therefore, successful retrieval of webpages or websites depends on the data availability in the local network.

7DS also enables e-mail exchange among mobile users by implementing a Mail Transport Agent (MTA) that receives the e-mail from the 7DS clients and broadcasts it to the other mobile peers. When the peers reach the Internet (for example, when they visit a WiFi access point), 7DS forwards the e-mails to an SMTP server that delivers the e-mail to its intended destination.

3.1.2 Data sharing applications

In order to provide data accessibility for mobile users in disruption-tolerant environments, 7DS offers two important data sharing applications, naming file synchronization and bulletin board system (BBS). These applications, using a different approach than 7DS's web and email applications, try to maximize the information availability for mobile users in infrastructure-less networks.

3.1.2.1 File sharing and synchronization

The file synchronization application assists mobile DTN users in different scenarios. For instance, mobile team members who are creating and editing project documents in a col-

laborative fashion in a mobile environment are able to share their modifications with each other. Mobile users might create new files such as documents, graphs, and pictures while disconnected from a global Internet connection. They might also download and cache some data content when they go back online. To maximize data availability in a disruption-tolerant environment, mobile users should be able to share these data objects with other mobile peers in the absence of the Internet. Epidemic replication [Vahdat and Becker, 2000], which has been proposed as a technique to distribute data among hosts in mobile DTNs, can be used to provide all mobile users with copies of each data file. However, due to possible modifications and edits of these replicated files in each host, inconsistency among the distributed copies happens over time. Using epidemic replication as the only distribution method causes all mobile hosts to be flooded with new versions at each file update. Therefore, a file synchronization mechanism is required to reconcile the inconsistent files without this flooding, when mobile nodes are in physical proximity. Ideally, all hosts should have the most up-to-date copy of the file.

The file sharing problem has been studied in the context of maintaining large replicated collections of files or documents in connected distributed environments [CVS, 1990], [RSYNC, 2005]. Systems like CVS [CVS, 1990] manage multiple revisions of the same file by storing the current versions on a central server. Lacking central servers in infrastructure-less mobile environments turns file sharing into a challenging problem. In contrast to an asymmetric server-based model, in which files are added to and downloaded from a central repository, in mobile DTNS versioning and synchronization should be managed in an entirely P2P fashion. Every host should be able to add files to other hosts or download from those hosts directly through symmetrical uploads and downloads. Furthermore, due to intermittent connections with unpredictable lifetimes, communication costs to exchange files should be minimized to save link bandwidth. For example, in synchronizing large files, transmitting the entire file for each incremental update is a waste of bandwidth and time. Therefore, when new versions of a file are created on different hosts, 7DS syncs all copies to the most recently available version by transmitting only incremental updates or deltas as explained in Section 3.2.7.2.

3.1.2.2 News sharing in a BBS

File synchronization is classified as a pull-based data sharing model. In the pull-based file synchronization model, mobile applications are able to download new files from newly discovered mobile peers automatically. These uploads and downloads are transparent to the user. In a push-based model however, information is shared based on a user's preferences. 7DS BBS for news sharing falls into this latter category, in which mobile users decide whether to download the files after reviewing an advertised list.

Sharing information and news through an ad-hoc BBS are made clear by considering some example scenarios. For instance, students walking into a classroom might have visited various locations on the campus and have collected advertisements about upcoming events. By using a mobile ad-hoc BBS students are able to inform each other about different events on campus. In another scenario, customers strolling in a shopping mall might collect product advertisements and discount coupons from different vendors and download them into their handheld devices. These users are able to distribute these advertisements and coupons from device to device among other shoppers using 7DS BBS.

BBSs in connected environments are implemented by dedicated central message repositories. By browsing these repositories, data consumers are able to locate and retrieve the data posted by data providers. However, using central servers to post and review messages is not feasible in self-organized, disconnected mobile networks. 7DS BBS enables mobile users to post and review community news in self-organized disconnected mobile networks in a P2P fashion.

Most bulletin boards serve specific interest groups. After reviewing the metadata, mobile users choose to download information based on its significance. Unlike automatic updates in the pull-based file sharing model, BBS gives more freedom to the users to download just information that interests them. In the BBS shown in Figure 3.1, mobile users can review event advertisements at step 2, after their publication by the publisher at step 1 without any pre-configured infrastructure.

An alternative design involves building a BBS by installing stationary message repositories equipped with wireless interface in some designated areas. These info-station-based model repositories [Goodman *et al.*, 1997; Ganchev *et al.*, 2010] provide high- bandwidth

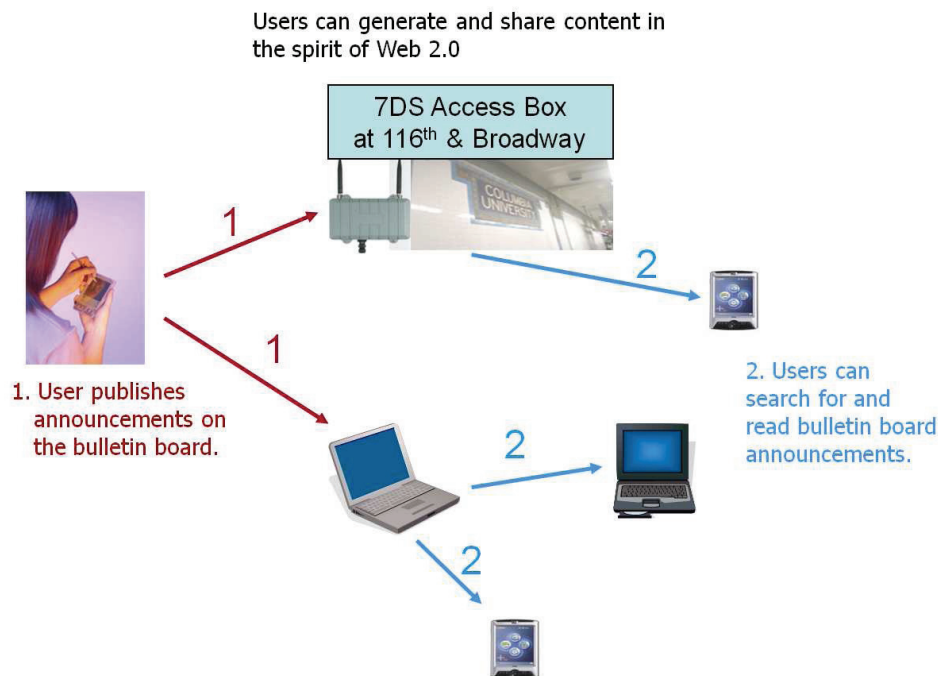


Figure 3.1: After publishing announcements at step 1 using BBS, mobile users can review and download the related information at step 2.

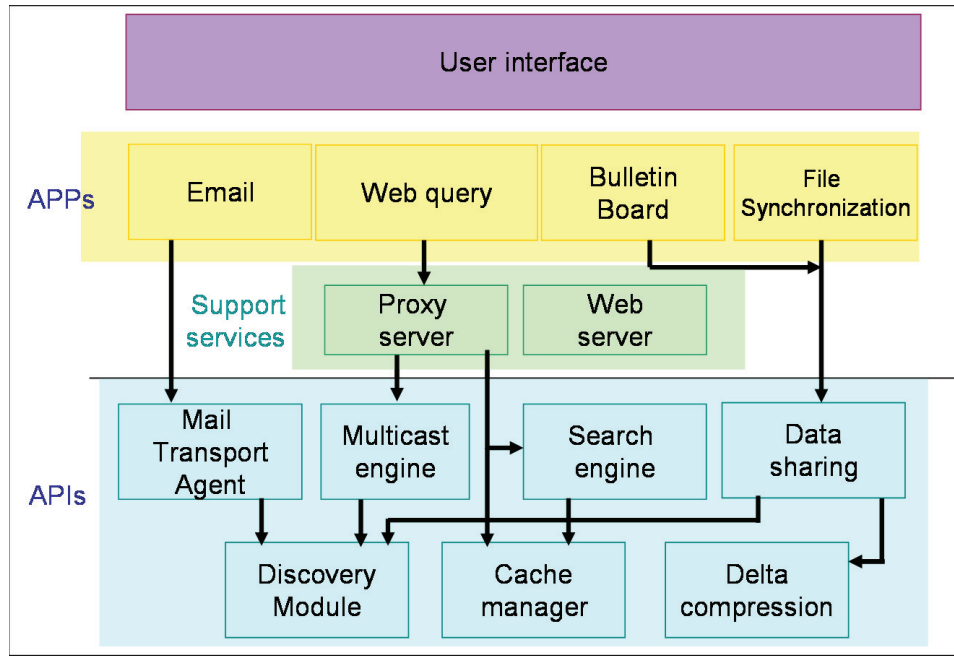


Figure 3.2: Modular architecture of the of the 7DS platform.

wireless connectivity in isolated coverage areas. They advertise information to the mobile users as they pass through their wireless coverage areas. Mobile users communicate with these info-stations to review or post messages. However, in the 7DS BBS model, mobile hosts communicate directly with each other in an entirely P2P fashion.

3.2 Architecture and characteristics of the 7DS platform

7DS has been developed as a platform to provide the applications and necessary core components for communication in disruption-tolerant environments. This platform constructs an abstract layer to conceal networking details and disconnections from the applications. We have built the 7DS platform, as shown in Figure 3.2, with a modular architecture. This will allow developers to expand the system by developing necessary plug-ins or modules for future applications. In the application layer, 7DS provides web browsing, email, file synchronization, and a BBS, each with the corresponding user interface to use the application. These applications are implemented on top of more generic modules which provide different core functionalities through their service APIs. 7DS has been implemented and

released for public use [Moghadam *et al.*, 2006] as a bundle of applications and underlying modules which are implemented in C and Java for both Linux and Windows platforms. The necessary components of the 7DS system for web query and email are proxy server, web server, search engine, multicast, and mail transport engines [Srinivasan *et al.*, 2007]. The 7DS architecture has been further expanded by adding file synchronization and BBS applications and their necessary components: data sharing and delta compression [Moghadam *et al.*, 2008]. A discovery module is an essential networking component which enables 7DS nodes to discover and set up a local communication network [Moghadam *et al.*, 2008] in the absence of a DHCP server [Droms, 1997]. Modules of the 7DS platform will be explained in more details in the following sections.

3.2.1 Discovery module

The discovery module is built on top of a multicast Domain Name System (mDNS) [Steinberg and Cheshire, 2010] protocol. The discovery module uses the mDNS protocol to enable mobile devices to discover each other and to assign each of them a unique IP address to create a local network automatically, without any server such as DHCP [Droms, 1997] or DNS [Mockapetris, 1987]. Furthermore, it helps 7DS users to announce their own services or locate the presence of the mobile services offered by other 7DS mobile users.

Each mobile user advertises both the mobile services that it is offering—such as web, email, file synchronization, or BBS application and the location of the service (IP address and port number) through the discovery module. After learning about these services, applications willing to exploit them resolve each service to its corresponding location. The discovery module then can use DNS Service Discovery (DNS-SD) to browse the network for services offered by local hosts. It next performs DNS queries over IP Multicast to look up the locations of those services and to resolve the name of a service to a location.

7DS mobile applications communicate among themselves in the local mobile network through publishing and subscribing to services. In 7DS every type of information such as file versions, application names, and their locations is treated as a service. Applications developed on top of the 7DS platform are able to announce their own services or resolve the ones advertised by the other mobile users through the discovery module's APIs. The

7DS discovery module publishes a service description, which could be as simple as a service name, to all other mobile nodes that are listening to the published messages. Mobile users who are interested in the advertised service use mDNS to map the service to the location of the corresponding host which offers the service. The discovery module also acts as an mDNS subscriber to receive service advertisement messages being published by the others in the network. As services are removed and added, their availability is stored in the memory of the mobile hosts. Based on mDNS protocol's standards, the discovery module caches resource records of the services in the mobile devices' memories. IP multicast looks up service names and resolves them to a location through this cached information, if the corresponding record has not expired yet. This enables the system to find services and their locations without a DNS server and through a P2P lookup. In order to use the discovery module in the application development, applications must specify and implement a callback function and pass it on to the discovery module API as an argument. The discovery module uses this argument to execute the right service routine for each corresponding service.

We have used Apple Bonjour [Bonjour, 2002] as an implementation of the mDNS [Steinberg and Cheshire, 2010] protocol to implement the discovery module. We have implemented a thin wrapper around DNS-SD [Bonjour, 2002] service advertisement and discovery APIs to hide its complexities from the application developer. Our Java discovery APIs [Moghadam *et al.*, 2006] are simpler than Apple Bonjour but provide sufficient service resolution functionality for disconnected environments.

3.2.2 Proxy server

The 7DS proxy server emulates a connected communication path for the web browsing and web query applications in the absence of the Internet. In the absence of a connection to the Internet, the 7DS proxy server thus facilitates the exchange of requested information among mobile peers. It uses the search engine (Section 3.2.4) and the multicast query engine (Section 3.2.5) to search the shared files and web-pages cached in the mobile devices to locate the relevant information. If the URL requested by the mobile user exists within the local network, the proxy server satisfies the mobile user's request by providing the exact web-page. On the other hand, if no matching web-page to a URL can be located, the

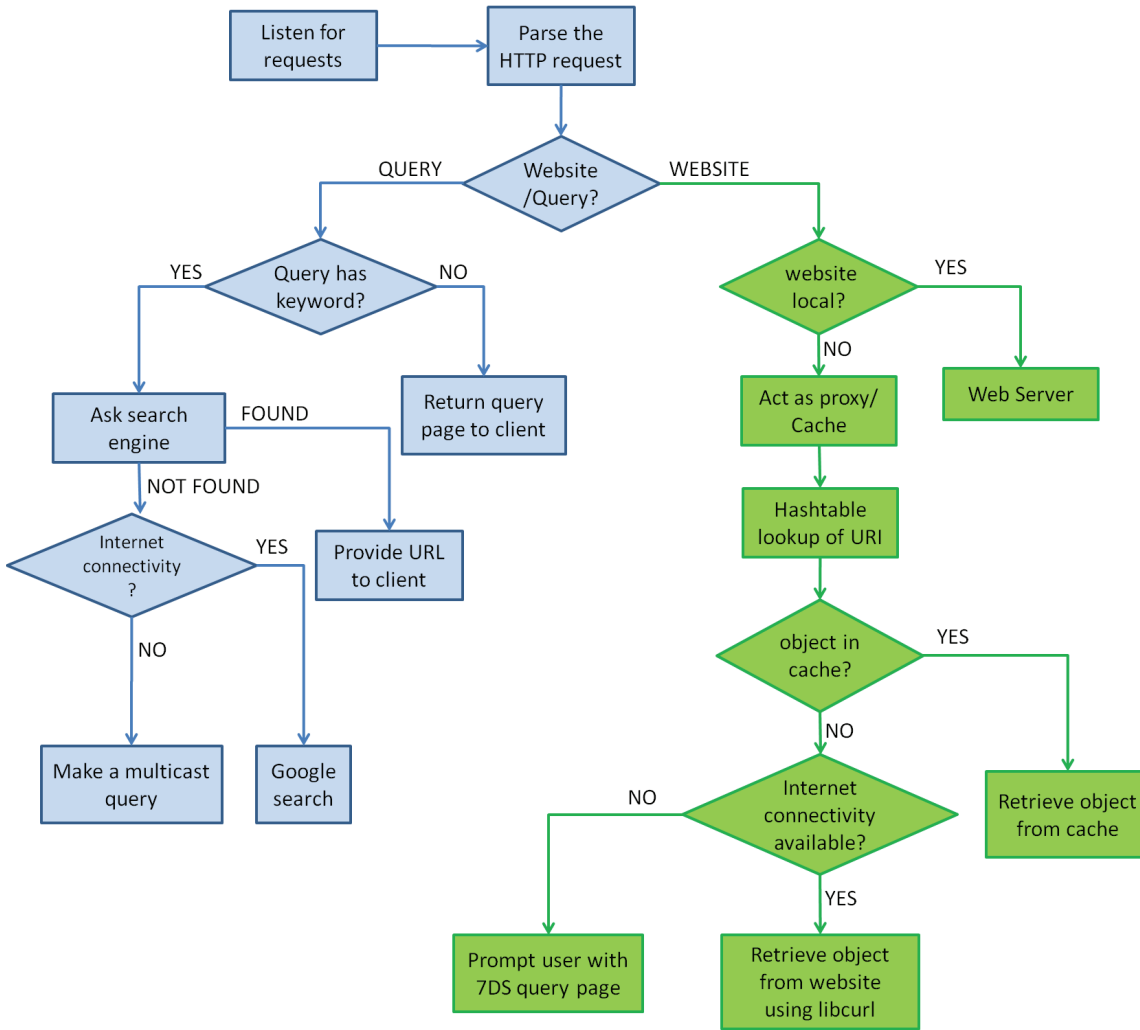


Figure 3.3: The algorithm of the 7DS proxy server for handling HTTP requests

proxy server uses the multicast query engine to search for web-pages relevant to the mobile users' search keywords. Thus the proxy server works as the interface between the user, the Internet, and other 7DS peers. In the absence of the Internet, the proxy server allows the user to exchange information by querying other nodes and searching their shared folders.

The proxy server listens to incoming HTTP requests; based on the type of request and whether the device is connected to the Internet, the proxy server decides to serve the request from the local cache, or the Internet, or through querying other 7DS system nodes via the 7DS multicast engine.

Based on the incoming query, the proxy server retrieves the data object most relevant

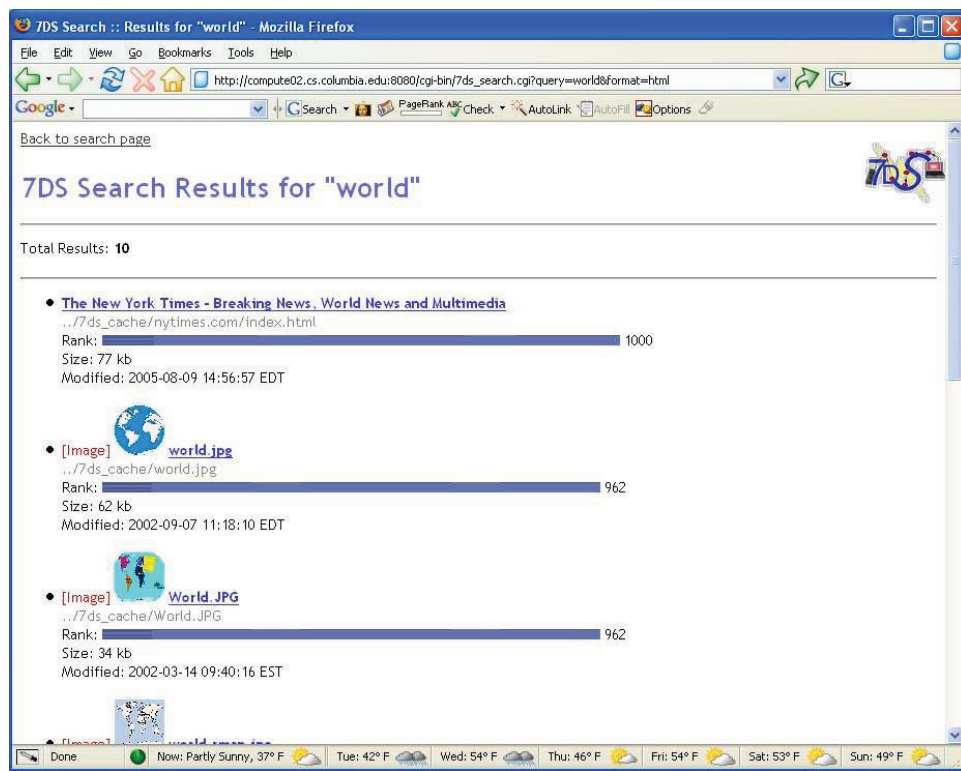


Figure 3.4: The 7DS search page shows results for a keyword search. The results correspond to matching files in the local cache.

to the user's request from the Internet or local cache, in that order and creates a separate process thread to serve each incoming request from the 7DS's web clients. The work-flow of the 7DS proxy server is represented in Figure 3.3. As this figure illustrates, the 7DS proxy server listens for the incoming HTTP requests from the mobile web client. If the request is for a URL, the proxy server first tries to retrieve the object from the user's local web cache. If the exact object is not in the cache and an Internet connection is not available either, the proxy server prompts the user with the 7DS query web-page, which asks for keywords that the user might wish to use to search for other related web objects or documents. After the user's HTTP query for certain keywords has been received, those keywords are multicast by the proxy server within the local network through XML messages. The 7DS proxy server then retrieves the responses from the other mobile users, orders them based on their relevance, and presents them to the user in a format represented in Figure 3.4. The URLs relevant to the corresponding query are displayed to the user in order of relevance, based on the frequency of the search keywords appearing in the documents or web-pages.

3.2.3 Local web server

The web server on the 7DS system serves two functions. First, it runs the web-based user interface to the 7DS system. Second, it works together with the proxy server to display local cached results in the absence of an Internet connectivity.

The web server needs to run on small mobile devices and sometimes embedded devices. Therefore, it needs to have small memory and disk footprints. One such small, open-source web server that we have used in the 7DS system is `thttpd` [THHTTPD, 2005]. However, any web server that supports CGI and PHP can be used in conjunction with the 7DS system, as the 7DS system uses a folder where shared web objects are placed. This shared directory can be searched and indexed by the other 7DS system components such as search engine and cache manager, as explained in the next section. The accepted queries from other mobile users in the local network are fulfilled from the web-pages and documents stored in this directory.

3.2.4 Search engine and cache manager

The search engine and the cache manager assist 7DS nodes in the query process. The search engine provides the user with the ability to find the corresponding files to the requested keywords or URLs that exist in the device's internal cache, and the search engine locates these files by using the indexes generated by the cache manager. Running as a daemon in the background, the cache manager creates an index of the files that reside in the cache. It checks frequently-at intervals of 20 seconds-whether there have been any updates to the cache and updates the files' index accordingly. If there have been no updates to the files in that directory, then the cache manager just goes to sleep without taking any action.

The search engine is a CGI binary that runs on the local web server and searches the index created by the cache manager.

3.2.5 Multicast query engine

The multicast query engine helps exchange information among peers in the network to search and locate relevant information using search keywords. The user first enters search keywords through the 7DS's web interface. This query is then added to the device's internal database. The query scheduler broadcast engine broadcasts the query list in an XML-encoded string to the network and the query engine reads the list of queries, encodes them in an XML-formatted string, and broadcasts the string in a multicast packet. The query scheduler sleeps for a small interval (20 seconds by default) and then resumes and broadcasts again.

The query receiver listens for the incoming queries and, upon receiving a query list, runs a local search on the device using the search engine. If related information is present, the query receiver encodes the data in a RSS-based XML format [RSS,] and sends the XML as a response of the query in UDP packets to the requesting peer. The query response is sent in unicast fashion.

The user is then presented with a dynamic page that lists all the results corresponding to the user's query. This dynamic page, which is generated by a CGI binary, refreshes every 10 seconds and provides the user with an updated result list. In order to avoid duplicates, the queries, the results and their corresponding peers are stored in a SQLite query database [SQLite, 2005], which is an open-source database engine with a small-footprint. Unlike the

larger database engines such as Oracle, SQLite does not require a daemon to handle SQL requests and is hence very suitable for embedded devices with limited CPU and battery resources.

3.2.6 Mail transport agent

In addition to functioning as a query/response system, 7DS is also designed to perform data gathering and data delivery. The core communication protocol for this part is SMTP [Klensin, 2001].

The SMTP server listens to the incoming email messages and transfers those which are intended for other users and which should be propagated through the network to the local Message Transfer Agent (MTA). The MTA later sends these messages to its neighboring MTAs. MTA also takes care of managing and storing all the received e-mails in each 7DS mobile node.

The SMTP server receives emails from the email client and creates a SHA1 hash of the email and recipient information. The tuples of email hash and sender information are stored in a database as a measure against receiving duplicate messages or sending the same email to its own original sender, should the sender be encountered again in the future. When a 7DS node meets another node, its MTA goes through the hash-table and the email directory, reads all the stored emails, and sends them to the peer's MTA. When the node visits a WiFi AP and, therefore, Internet connection becomes available, the transport engine sends all the stored emails to the Internet. Because of problems with e-mail duplication, which could happen if two different carriers of the same message meet some AP, our implementation filters the e-mails through a single server before submitting them to the Internet. The main software library that we have used to implement SMTP functionality is libESMTP [Stafford, 2002].

3.2.7 Data sharing module

Protocols that enable mobile peers to share and synchronize data have been bundled in the data sharing module. As explained in Section 3.2, the data sharing module of the 7DS platform is composed of two sub modules, BBS and the pull-based file synchronization. The

data sharing module implements the necessary components for file synchronization and BBS applications of the 7DS system.

Since mobile users might individually modify the shared files and data objects, inconsistency among the shared copies might arise over time. The file synchronization application of 7DS helps mobile users to reach a consistent state by reconciling these inconsistencies. The version controller, as part of the data sharing component, determines the inconsistencies in the shared data among mobile nodes. The data sharing component, then, resolves these inconsistencies by applying a predefined policy. Delta encoding is another important part of the data-sharing component which helps save communication bandwidth, especially when dissimilarities between different versions of the shared file among mobile hosts are not significant. In this case, the delta encoding component saves communication bandwidth by just transmitting the partial differences or deltas that are used to build the new version of the file on the hosts which have stale versions. Since the data-sharing module provides the necessary functionality for the file synchronization and BBS applications, we explain this module and its compartments by referring to their usage in these applications.

File Synchronization File synchronization implements the pull-based data-sharing model. In the pull-based model, mobile peers in physical proximity synchronize their shared objects automatically and transparently from the application by pulling the latest version. Mobile hosts willing to share data objects with others define a shared directory in which all shared objects are placed. Applications commit their file modifications into this directory. The reconciliation policy is to update the shared directory in all peers to the latest version of data objects through automatic version discovery and data downloads. The version controller component of the data sharing module keeps track of multiple file versions which are created over the network. All file versions are announced throughout the network via the discovery module. After discovering new data objects by the version controller, the file-sync component uses the delta-encoding module to retrieve the most up-to-date versions in a bandwidth-efficient way. We will explain the details of the version controller and delta-encoding sub modules in Sections 3.2.7.2 and 3.2.7.1.

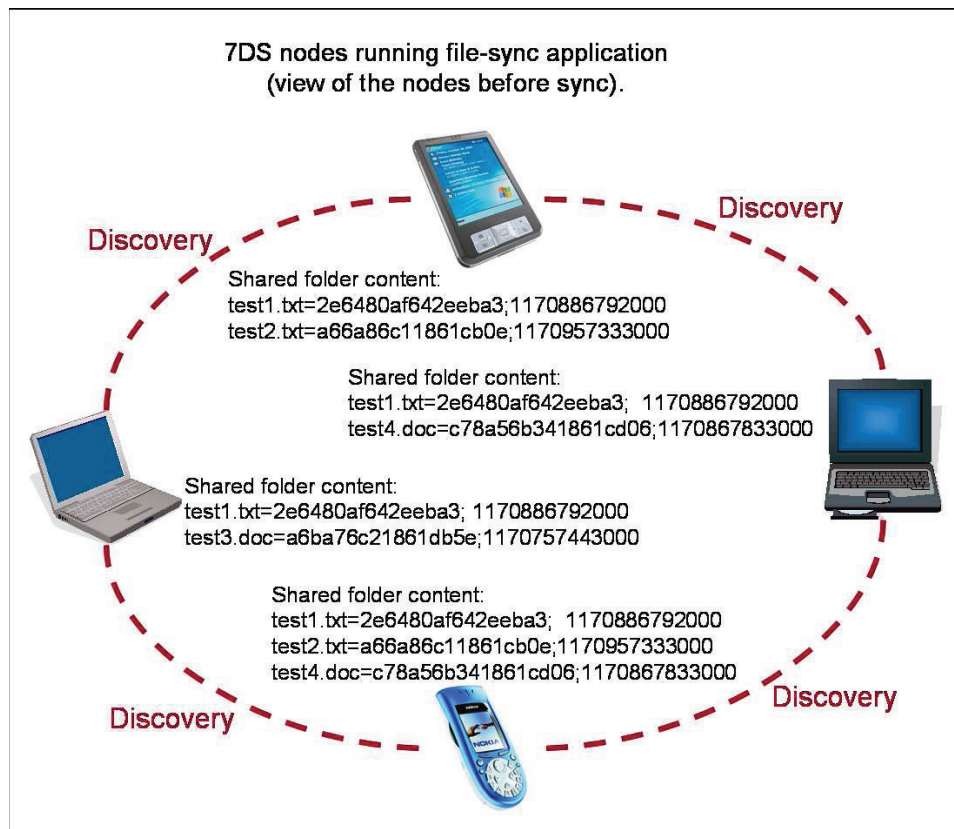


Figure 3.5: Mobile nodes discovering other peers and file information lists offered by them. Each file information embodies a hash value and date tuple.

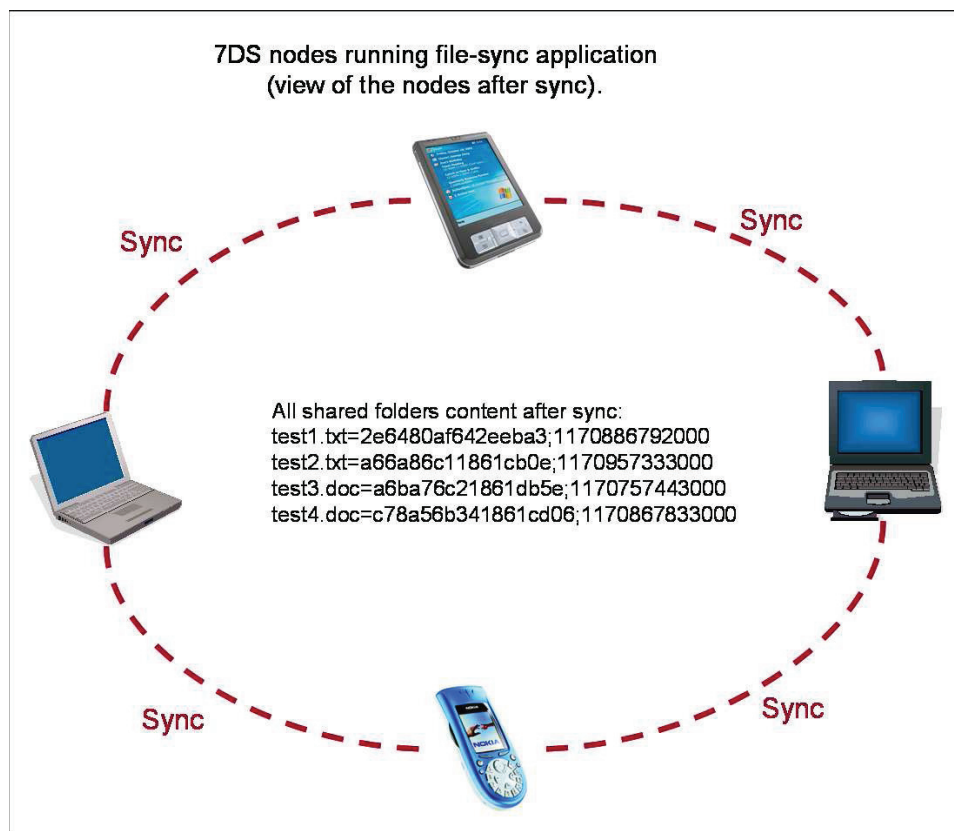


Figure 3.6: Mobile nodes' shared folders after file synchronization.

Bulletin Board System The BBS is responsible for the exchange of data in the push-based mode in which the mobile node receives an explicit request for data before any actual data transfer happens. Here, unlike file-sync, data exchange does not take place automatically and transparently to the application. The push-based model is suitable for distributing information based on users' interests. In this model, metadata which is a list of advertised data objects is exchanged after discovering new mobile users. Then, based on the user's interest in the advertised data, a download request might be generated. Similarly to the pull-based data sharing, all advertisements are published and discovered through the discovery module. The implementation of the BBS system is based on exchanging advertisements in XML format among mobile users. In the BBS system, mobile users decide whether or not to send an explicit download request for a specific data object after reviewing the advertisement. If a user decides to download the actual data, the file transfer happens in a unicast communication fashion. The pull-based file-sync module is more complicated, and its components and implementation are described here in more detail.

3.2.7.1 Version controller for the pull-based file sharing application

The version controller, which is an event-based process, runs in the background and listens to Jnotify's [JNotify, 2002] notifications about changes in the shared directory upon any insertion or deletion. Jnotify [JNotify, 2002] is a Java wrapper on top of inotify which is a Linux kernel subsystem that acts to extend file systems to notice changes to the files automatically. The version controller determines all modifications in the shared directory by registering for inotify's notifications. After a thorough scan of the shared directory, the version controller announces these versions by multicasting tuples of hash values and modification dates through the discovery module. Upon the discovery of new nodes, file information lists are exchanged with other mobile nodes, as shown in Figure 3.5. The version controller residing on each mobile host compares its own file information with the lists advertised by other peers in the network. The shared directories which are missing a file or containing some older versions are reconciled automatically to the most up-to-date versions. After discovering the new versions, the synchronization requests are issued by the file-sync module on behalf of the mobile application. After reconciliation all participating

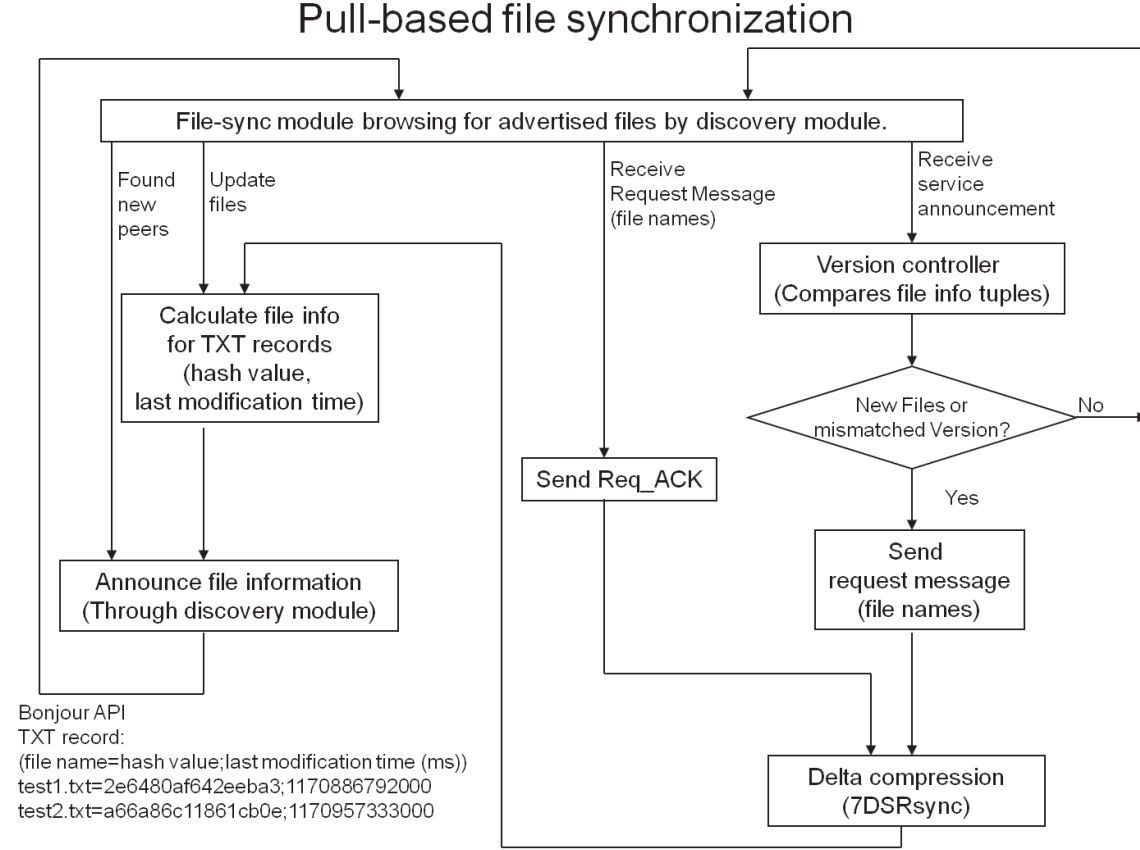


Figure 3.7: File synchronization algorithm

nodes have a globally consistent view of the shared folder as shown in Figure 3.6.

Pull-based file synchronization work flow Pull-based file synchronization module as represented in Figure 3.7, employs the discovery module to periodically announce file information lists as tuples of hash values and modification dates of the files residing in the shared folder. After discovering new data objects by the version controller and through discovery module, file-sync component exploits the Rsync delta encoding protocol to retrieve the most up-to-date versions in a bandwidth efficient-way. 7DSRsync is an implementation of the Rsync [RSYNC, 2005] delta encoding algorithm for the 7DS platform. 7DSRsync implements the Rsync protocol steps required for communication between the client and the server of the rsync algorithm. We used Jarsync [Tridgell, 2000], which offers the primary libraries for the rolling and MD4 checksums of the rsync algorithm to implement 7DSRsync

module. Jarsync library [Tridgell, 2000] is an effort to implement an equivalent of librsync [Librsync, 2005] for Java platforms. It also provides necessary APIs to create deltas or file differences by scanning the new file residing on the server. But unlike librsync [Librsync, 2005], the client and server of the Jarsync library have not been implemented [Tridgell, 2000].

The 7DS file synchronization application uses the JNotify library [JNotify, 2002] to listen to file system events such as file creation, modification or deletion. JNotify works on both Windows (Windows 2000, XP, Vista) and Linux with INotify support (Kernel 2.6.14 and above). The JNotify Linux API is a thin wrapper around the Linux INotify API. Due to the complexity of the Windows APIs for file system events, most of event handling logic has been put into the C++ code for Windows JNotify. After every insertion, deletion or file modification in the shared folder the new file info list is announced through the discovery module which encompasses BonAHA APIs.

3.2.7.2 Delta encoding for the pull-based file synchronization application

In disruption-tolerant environments, wireless contacts among mobile users are intermittent with unpredictable lifetimes. In order to avoid disruptions in data transmissions, delta encoding module is developed and added to the 7DS architecture to implement a more bandwidth efficient file synchronization and data exchange application in these types of communication environments with intermittent connectivities. Usually most of the update requests cause the retrieval of slightly modified instances of a resource for which the client already has a stale entry. The Rsync delta encoding algorithm [RSYNC, 2005] is a way of efficiently transmitting a file across a communications link when the receiving host already has a stale version of the same file. This makes a more efficient use of network bandwidth by transferring a brief description of changes, instead of the entire new instance of the resource.

In the Rsync delta encoding algorithm, the requesting peer acts as the client in the protocol and splits its old version of the file into a series of non-overlapping blocks of size s , as shown in Figure 3.8. For each of these blocks, the client with the old file calculates two checksums: a weak *rolling* 32-bit checksum and a strong 128-bit *MD4* checksum. The client sends these checksums to the host with the new version of the file. The host with

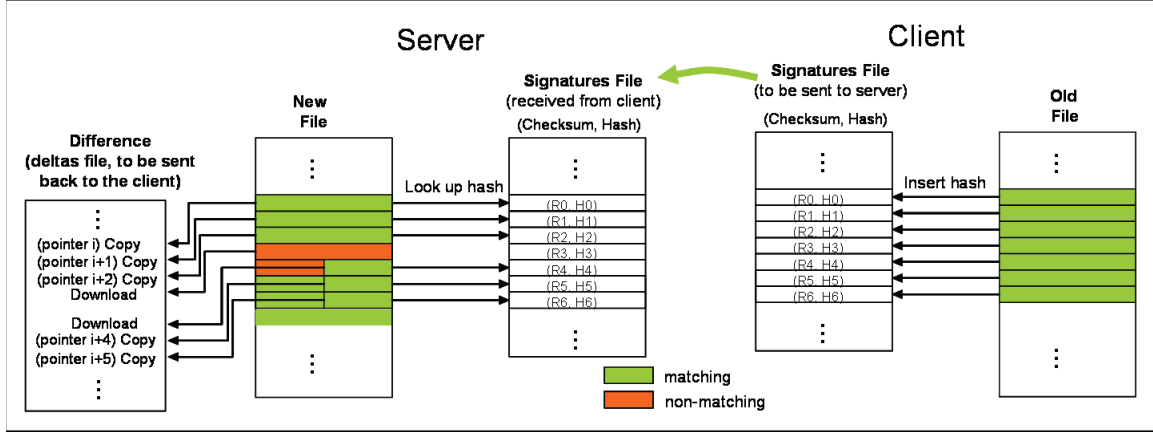


Figure 3.8: Client generates signature file and server creates delta file according to the Rsync delta encoding algorithm.

the new version operates as the server in the Rsync algorithm. The server scans through its own new version of the file looking for all matching blocks of size s with the same weak and strong checksums. (The non-matching blocks can be discovered with a high precision by only using the weak rolling checksum which is cheaper to calculate than the strong MD4 checksum). The scan starts at different offsets (not just multiple of s), so the modifications in the middle of the file won't cause the entire file to be transmitted. After this step, the server sends the client a sequence of instructions-called deltas-for constructing a copy of the new file. Each instruction is either a reference to a block of the old file residing on the client or the literal data. Literal data is sent for only those sections of the new file which did not match any of the blocks of the old file. The end result is that the peer with the old version of the file constructs a copy of the new file.

Only pieces of the new file that are missing at the client plus a small amount of data for checksums and block indexes are sent over the link. This saves link bandwidth by avoiding unnecessary transmissions especially when the file is large. Moreover, the algorithm only requires one roundtrip to exchange checksums and deltas.

3.3 Scalability of the 7DS platform

Since we haven't tested the 7DS system and its applications with real users, in this section we discuss the scalability of 7DS from an analytical perspective. All 7DS applications introduced so far, are responsible for data propagation in different mobile scenarios. Among these, the file synchronization module is in charge of transferring large amounts of data sometimes, especially when exchanging audio and video files. Scalability of this module of the 7DS platform should be investigated in more detail. The mobile nature of the network makes duration of the link connectivity and number of concurrently connected mobile hosts completely unpredictable. In dense and highly mobile networking scenarios, a large number of mobile nodes meet simultaneously for short periods of time. File exchange among these hosts imposes a large amount of traffic on the network for this time interval, especially when all nodes have some new files to share. Therefore, the demand on each host increases in proportion to the total number of hosts, quickly overrunning the network's limited capacity.

Suppose there are N mobile nodes in the network, each of which has a new file with size M MB to share. Distributing a totally new file is expensive in terms of bandwidth consumption. Suppose the links bandwidth is B Mb/s. After the discovery phase, which takes place using multicast communication, all N mobile nodes contact each other via unicast to download the new file. Therefore, the transmission bandwidth to all these nodes becomes $\frac{B}{N}$. As a result, all nodes requesting the file receive it after $\frac{8MN}{B}$ sec. Given this, connections between the file owner and receiving mobile nodes should at least last for $\tau = \frac{8MN}{B}$ sec. If the link lifetime, before mobile nodes go away, is T , we should have: $\frac{8MN}{B} < T$ for a successful file transfer. After rearranging this equation, we get: $N < \frac{TB}{8M}$. Therefore, scalability of the system is bounded by the speed of mobile nodes, size of data objects, and bandwidth of the links.

3.4 Related work

Several systems have been developed to facilitate communication and application development for mobile networks. Some of them exclusively target always connected networks. Therefore, these systems fail or operate poorly in mobile DTNs with temporarily available

wireless links. Moreover, several of these systems have been specifically proposed to address only the file sharing problem. 7DS has been designed and developed considering limitations of mobile DTNs. Furthermore, 7DS, as well as offering disruption-tolerant applications, provides a comprehensive platform to develop new applications.

Some of the systems in this research area have been developed to address the problem of Internet connectivity in the absence of a global network connection especially in undeveloped areas such as those addressed in the one-laptop-per-child project, OLPC [OLPC, 2005]. The goal of the OLPC [OLPC, 2005] networking project is to connect remote wireless users who do not have any Internet infrastructure to the Internet. Connectivity is achieved by using mesh networking technology. OLPC utilizes conventional ad-hoc protocols such as AODV [Perkins *et al.*, 2003] or OLSR [Clausen and Jacquet, 2003] for routing between mobile laptops. In contrast, 7DS is designed to operate in sparse mobile disconnected networks via store-carry-forward routing.

In terms of file sharing, applications such as Gnutella [Gnutella, 2000] and BitTorrent [BitTorrent, 2001] come to mind first. However, they have been designed to search for data objects in always-connected environments. Some systems like Hayes's [Hayes and Wilson, 2004] have been developed to share files on a mobile ad-hoc network based on Gnutella protocol. However, these systems suffer from extra overhead that Gnutella's routing and file exchange protocols create for mobile networks. Furthermore, in contrast to Hayes's system [Hayes and Wilson, 2004] that runs only on Bluetooth, 7DS is capable of operating on any type of network.

Microsoft Groove [Groove, 2007] is a commercial shared workspace in which groups are built via invitations. So, unlike 7DS, there is no notion of node and service discovery. Instead, all changes are sent to all users or to a dedicated server after file modifications.

Klemm *et al.* [Klemm *et al.*, 2003] have built a P2P file-sharing system called ORION (Optimized Routing Independent Overlay Network) for mobile ad-hoc networks. This system uses an overlay network that combines application-level query processing with network layer route discovery for file sharing. 7DS' multicast system works similarly, but it does not require a routing system. Furthermore, 7DS, in addition to solving the file sharing problem, provides an application development environment for mobile DTNs.

Cache-and-forward (CNF) protocol [Paul *et al.*, 2008] is implemented as an overlay network for content distribution among mobile nodes that are disconnected from a global Internet. CNF distributes data through the hop-by-hop transfer of large data files between stationary CNF routers, using a reliable end-to-end transport protocol like TCP. Therefore, CNF implements a hybrid network of stationary data repositories which participate in the routing of data files among mobile users, while 7DS implements an entirely mobile peer-to-peer communication model.

Dropbox [Dropbox, 2008] is a cloud-storage service for sharing files among different computing devices. In this application the shared files are hosted over the cloud, and the shared folder can be viewed across different devices such as desktops, laptops, and Smartphones. Dropbox, however, is specifically designed for always-connected networks on the assumption of an always-on server cloud.

iClouds [Heinemann *et al.*, 2003] and Clique [Nioclais *et al.*, 2003] were developed for file sharing in disruption-tolerant environments. Clique synchronizes files among devices which are connected to the same communication channel. Its inter-node communication is based on broadcasting hash values of fixed-size data chunks to all participating nodes. Considering just fixed-size file chunks might result in sending the entire file in the case when a minor insertion has been made in the middle of the file. This overhead plus the overhead caused by unnecessary broadcasts create extra traffic and utilize a large portion of the mobile network's capacity. Unlike 7DS, which uses mDNS as its core discovery protocol, iClouds [Heinemann *et al.*, 2003] uses a UDP ping/pong mechanism to discover nearby nodes. Data versions in iCloud are matched by comparing the actual data represented in XML format through string matching. However, in 7DS, comparing hash values instead of the real data results in a more efficient version control system.

Some projects have targeted the problem of application development for mobile ad-hoc networks. Proem [Kortuem *et al.*, 2001] and Peer2Me [Wang *et al.*, 2007] are two platforms for developing mobile ad-hoc applications. They both use Bluetooth device discovery protocol to search for all nearby Bluetooth devices. After the device discovery phase, mobile nodes exchange their profiles to announce the services that they have to offer. Neither Proem nor Peer2Me supports the notion of the automatic service discovery as the 7DS platform

does. Proem [Kortuem *et al.*, 2001] offers a protocol stack to provide naming, discovery, communication, and security in ad-hoc networks and offers an MP3 file sharing application developed using this middleware. However, applications offered by the 7DS platform are more general and cover a broader range of communication requirements in mobile DTNs. Peer2Me nodes use a gossip mechanism to inform each other about the presence of other nodes and use the OBEX protocol for communication.

Haggle [Su *et al.*, 2007] offers an architecture that isolates the applications from the underlying communication mechanisms. In this architecture, applications delegate the task of handling and communicating data to Haggle, which in turn adapts to the current network environment using the best available connectivity. The Haggle prototype currently supports web and email applications.

JXTA [JXTA, 2001] is a library that enables development of XML-based P2P protocols to allow peers in a network to interact with each other. However, just like the Gnutella and BitTorrent networks, JXTA is suitable for devices that are always connected to the Internet.

3.5 Conclusion

In the absence of a ubiquitous connectivity such as the Internet, the 7DS system presents a good solution for implementing transparent data exchange in mobile DTNs. As devices join and leave the local mobile network, they bring in new information or carry out information that needs to be sent to the global network. The components of the 7DS system enable emails and webpages to be exchanged within the mobile DTN easily. Furthermore, 7DS provides a platform that serves as an environment to develop mobile disruption-tolerant applications. The modular design of 7DS facilitates future expansion of the platform, independent of the underlying core system. We have introduced a new class of applications and specified their fundamental requirements in DTNs. The main significance of these applications lies in their enfolding a broad range of core requirements in device and service discovery, network setup, data exchange, and efficient bandwidth usage. Considering these requirements, we designed and implemented 7DS's primary APIs for service reso-

lution, multicast query, messaging, search engine, delta encoding and, data sharing. We further implemented different data-sharing applications such as file synchronization and BBS that implement both pull-based and push-based file-sharing models. We also implemented disruption-tolerant web and email applications that utilize the store-carry-forward communication mechanism provided by the 7DS platform. Setting up 7DS on any device or computer is easy. 7DS-enabled devices can automatically discover each other, set up a local network, and exchange information to overcome the lack of the global connectivity. In the next chapter we will move from the application scope to the data routing in mobile DTNs. We will discuss open research problems in this area and we will design a new mechanism for data distribution based on mobile users' interests.

Part II

Content Distribution in Mobile DTNs

Chapter 4

Interest-Aware Content Distribution Protocol for Mobile DTNs

In the previous chapter we focused on mobile DTN applications and their requirements to operate successfully in disconnected mobile environments. We have presented 7DS as a mobile DTN application platform to assist mobile users with data discovery and exchange in a locally-created mobile network away from any global Internet connectivity. We also developed data sharing and email applications as part of 7DS applications for data dissemination in mobile DTNs. The key component for the data distribution between mobile users and applications, however, is the routing algorithm. Having an efficient routing algorithm is a key factor in enhancing users' experience in better utilizing the applications in mobile DTNs. An efficient routing algorithm is able to provide mobile users with their data of interest with the least amount of delay. This data delivery task is even more complicated in the absence of a connected path between source and destination. Routing algorithms in mobile DTNs handle data delivery from a mobile source to its destination to mitigate the absence of an end-to-end Internet connection. In this chapter and the next one we formalize the data routing problem in mobile DTNs, and we specifically focus on algorithms to distribute data among mobile users based on their interests. We introduce a new communication algorithm

to route data through the mobile DTN to mobile users who are interested in the content of the data. We further discuss how data dissemination applications for mobile DTNs could benefit from our interest-aware data routing algorithm to distribute files, news, and other data objects more efficiently.

Data dissemination in mobile DTNs spans four major categories based on the information flow from mobile nodes to the Internet, Internet to the mobile nodes, P2P communication among mobile nodes, or communication between stationary information centers and mobile nodes such as bulletin boards. In this chapter we focus on a P2P data dissemination model that takes advantage of opportunistic wireless contacts between mobile users to distribute data.

Most of the routing protocols that have been proposed for mobile DTNs address the data routing problem between a source and a specific destination [Lindgren *et al.*, 2003], [Zhao *et al.*, 2004], [Thakore and Biswas, 2005], [Jones *et al.*, 2005], [Boldrini *et al.*, 2007]. However, in many real-world scenarios it is necessary to extend data transmission from individual recipients to a group of recipients. Propagating news or advertisements, managing emergency situations, and sharing images or documents among members of a community are examples of communication with a group of recipients. Proposed models for multicast communication in mobile DTNs try to extend the classical IP multicast model to disruption-tolerant networks. They define new multicast semantics to overcome the intrinsic network partitioning in mobile DTNs [Zhao *et al.*, 2005], [Chen *et al.*, 2006]. However, these proposed architectures require a global knowledge of the multicast group memberships of mobile users and the network topology [Zhao *et al.*, 2005]. Therefore, implementing these architectures is infeasible in mobile DTNs which suffer from a frequently changing topology and lack of infrastructure to track group memberships. Hsu *et al.* [Hsu *et al.*, 2008] have proposed profile-cast as an architecture to distribute data among groups of mobile users that are defined based on their mobility profiles. These mobility profiles are defined as the history of the geographical locations visited by the mobile users. Profile cast relies on these mobility profiles to distribute a message among groups of mobile nodes with similar mobility patterns. But this model is oblivious to the fact that users with uncorrelated mobility patterns might actually have some common interests. For instance, different groups of tourists

visiting a city might be interested in the same data content even though they had visited entirely different locations in the past. In another scenario, passengers who are riding on a train, away from global wireless data coverage, might have completely different mobility histories but they might still want to locate data of their interest from other passengers who share the same interest.

Scenarios such as these would benefit from a more intelligent content distribution algorithm that is able to learn users' behaviors and their interests over time. This algorithm should be able to categorize mobile users in the network based on their interests and to distribute appropriate data among the interested audience in the network. In this chapter we propose just such an algorithm, one that automatically learns users' behaviors and extracts their preferences in order to assign them to appropriate interest categories. Our algorithm extracts users' interest-vectors from their cached data content and uses these vectors to route specific data to its interested recipients. When two mobile users meet, they exchange these interest-vectors in order to calculate each other's interest in the data which each of them is trying to distribute. Appropriate data is then transferred to the other user only if there is enough correlation between that user's interests and the content of the data which its partner has to offer. Therefore, unlike classical IP multicast, data is diffused throughout the network through the intermediate users who are also interested in the data content.

We have implemented our interest-aware content distribution algorithm in The ONE [ONE, 2007] simulator for mobile DTNs. We ran several experiments using the reality-mining database [Mining, 2003], which is one of the largest, real-world mobile phone traces collected in academia. Our experimental results show that, compared to epidemic routing, our algorithm significantly increases the amount of relevant data of interest received by the mobile users in the mobile DTN. This increased coverage is achieved without distributing irrelevant data among those users.

In Section 4.1, we motivate the necessity of an interest-aware content distribution protocol for mobile DTNs. Section 4.2 discusses related work in this area. We discuss the details of our interest-aware routing algorithm in Section 4.3, while Section 4.4 explains our implementation in the ONE simulator and our choice of mobility traces. Section 4.5 evaluates our simulation results in detail. Finally Section 4.6 concludes the chapter.

4.1 The necessity of an interest-aware distribution mechanism

There are many mobile DTN applications that benefit from the transmission of data to groups of recipients rather than to individuals. In some of these scenarios, a message needs to be transmitted to every mobile node located in the DTN. For example, in situations such as emergencies, traffic congestion notifications, or severe weather alerts, all mobile nodes present in the area would want to be notified. In these scenarios all mobile users can receive the data through epidemic routing. On the other hand, there are situations where not everyone needs to receive a specific message and the message or data content should be directed only to interested users. For example, information such as market news, sports events, scientific articles, or advertisements about particular products are of interest to only a limited subset of the total audience. In these applications, an intelligent data distribution algorithm must be employed to identify and select the interested recipients of this specific data content.

The problem of multicasting has been studied in mobile DTNs to handle communication with groups of users [Zhao *et al.*, 2005]. This multicast architecture is based on a group membership model, where mobile users register with a specific multicast group to exchange multicast messages with other group members. However, implementation of this model is infeasible in mobile DTNs, since there is no infrastructure to track group memberships.

Here, we introduce an interest-aware communication model that does not require users to obtain group memberships. Instead, our algorithm learns users' interests from their previously downloaded and cached data content and properly assigns them to distinct interest groups. Furthermore, our algorithm uses this inferred information to provide users with their favorite content efficiently.

Our interest-aware communication model falls between the unicast and epidemic multicast communication models. Unicast, involving communication between a single source and a single destination, requires to specifically address the recipient of a message. As we explained before, there are many DTN applications, however, which need to address a group of users without knowing their individual identities. Therefore, the unicast model is

not well suited for these types of applications. Epidemic routing, assuming unlimited communication resources, has been proven to yield optimal performance in terms of maximum coverage and minimum delay in mobile DTNs [Vahdat and Becker, 2000], [Jain *et al.*, 2004]. This optimal performance is achieved because, upon meeting, all mobile nodes exchange all their cached data with each other. Therefore, epidemic routing is far from satisfactory because it creates a lot of unwanted traffic. Moreover, in reality mobile devices are limited in their storage capabilities and communication bandwidth. Also, the contact times between mobile users might be very short, especially in highly mobile scenarios. Short contact times as well as limited bandwidth might disrupt the complete transmission of the cached content. Our proposed interest-aware algorithm, on the other hand, limits the transmitted data to only the content of interest for mobile users. In fact, our simulation results 4.5 using real-world data traces prove that this algorithm outperforms epidemic routing significantly in distributing more data relevant to mobile users' interests, while using the same amount of communication resources.

An example scenario of an interest-aware communication model is presented in Figure 4.1. In this scenario, two different communities of users are identified as X and Y . In this particular example, community Y are users who are interested in the data content D . The algorithm must recognize that the interested recipients of the data content D are users in group Y . Based on this algorithm, the sender (S) of data content D recognizes that node b has interests similar to those of community Y . Node b then transmits data D to other corresponding users who also belong to community Y . During this process the data is transmitted only through mobile users who are interested in its content.

4.2 Related work

After the invention of the epidemic routing more classes of routing protocols have been introduced for mobile DTNs. These successor algorithms try to use the communication resources more efficiently by avoiding flooding all mobile users [Jain *et al.*, 2004]. A major class of these routing algorithms which has been extensively studied, is based on a single-source, single-destination communication model. These algorithms try to locate a specific recipient

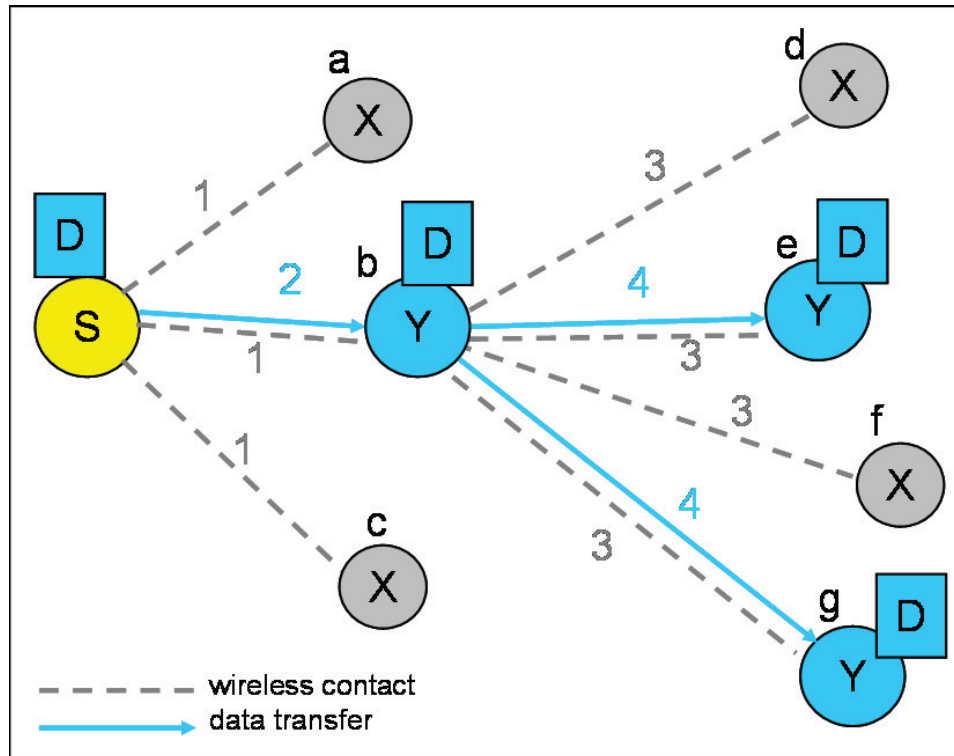


Figure 4.1: **Example of the interest-aware communication model.** 1 - The source (S) estimates the potential recipients of data content (D) as members of the community Y . 2 - After calculating the similarity between the interests of the encountered users and the data content D , the data is transferred to the most similar user to group Y which is user b . 3 - Node b meets new mobile users. 4 - After similarity calculations, two mobile nodes e and g are selected as the recipients interested in the data content D .

of a message through the mobile DTN. Some of these algorithms such as message-ferrying [Zhao *et al.*, 2004] rely on special mobile nodes called ferries to locate a certain recipient of a message. Other algorithms in this class use heuristics or probability measures to estimate the best path to the intended recipient. RPLM [Thakore and Biswas, 2005], practical routing [Jones *et al.*, 2005], and [Mundur *et al.*, 2006] base their routing decision on the history of the link connectivity. These protocols try to estimate the best path to the destination by monitoring link activations and deactivations over a time window. PROPHET [Lindgren *et al.*, 2003], another algorithm in this class, uses the history of encounters with other mobile nodes to update the delivery probability table at each node. The delivery probability table at each node indicates how likely it is that this node delivers a message to each known destination. All these single-source, single-destination algorithms need to specify the address of the recipient of a message. Our interest-aware routing algorithm, on the other hand, is designed to communicate with communities of recipients. It automatically discovers these communities based on the mobile users' interests.

Multicast communication with a group of recipients in mobile DTNs has been discussed in the previous research literature [Zhao *et al.*, 2005], [Chen *et al.*, 2006]. This multicast architecture requires tracking of the mobile users' group memberships. The multicast group memberships are managed by special nodes such as ferries in the message-ferrying approach [Chen *et al.*, 2006]. Designating special nodes to handle group memberships, however, is not a practical solution for mobile DTNs in general, because nodes are always joining and leaving the network. Our interest-aware architecture addresses the group-communication problem in mobile DTNs without explicitly tracking users' memberships. In our model, user communities are discovered on the fly, based upon their specific interests.

The following studies have addressed the content distribution problem in mobile DTNs. Leguay *et al.* [Leguay *et al.*, 2006] have tried to use a hybrid of mobile and stationary Bluetooth devices to maximize content diffusion among members of a group. However, the file synchronization application, developed as part of the 7DS architecture [Moghadam *et al.*, 2008], uses purely P2P opportunistic wireless contacts to distribute the most up-to-date data content among mobile users in the DTN. Both these architectures aim to maximize data dissemination epidemically without considering mobile users' interests, their current

states, or social contexts.

Similarly, context-aware file sharing [Conti *et al.*, 2007] is a file sharing application for mobile DTNs. This application tries to share files among users based on their interests. In this application users' interests are inferred from the extensions of the files which are stored in their mobile devices's caches. However, files with different extensions could still contain similar content, a situation that our interest-aware algorithm takes into consideration. Using a slightly different concept, HiBOp [Boldrini *et al.*, 2007] bases its routing decisions on the context and defines context to be the history of previously visited geographical locations. Knowing the current context of each mobile user, or in other words the mobile user's mobility history, and comparing it to the contexts of recently encountered users, HiBOp updates the history table of the mobile user. Using these history tables, this protocol calculates the probability of delivering a message to its destination node through each encountered intermediate relay. In using history of contacts in this way, HiBOp is very similar to the PROPHET protocol [Lindgren *et al.*, 2003] in using history of contacts. The use of this history in HiBOp, however, is further enhanced by exploiting users' contexts.

Two other context-based routing protocols, Profile-cast [Hsu *et al.*, 2008] and MobySpace, [Leguay *et al.*, 2005] consider mobility behavior of mobile users in their forwarding decisions. This mobility behavior is derived from the history of the geographical locations visited by the mobile users. These two protocols try to route a message closer to its destination by forwarding the message to the relays with similar mobility pattern to the destination. These algorithms limit the potential recipients to the mobile users who visit the same geographical locations as the destination. This, however, is not ideal. Mobile users might still share the same interests, even though they visit different locations or come from different neighborhoods.

Hui and Crowcroft [Hui and Crowcroft, 2007] take a different approach to show how data distribution can be improved by distributing data among relevant communities of mobile users. They define different communities of users by explicitly assigning their devices different labels. The labels are assigned according to mobile users' affiliations to predefined communities. They show by this experiment that this user labeling can significantly reduce the delivery cost in a single-source, single-destination routing model, without trading off

very much against the delivery ratio. Our interest-aware algorithm, however, automatically learns users' communities based on users' interests which are latent in their cached data.

4.3 Interest-aware routing algorithm

The first step in routing data content to mobile users by their interests is to construct subgroups of users by clustering them into appropriate interest communities. User and document clustering has been studied extensively in the context of web usage mining [Grear *et al.*, 2006]. In this context, personalized information delivery is achieved by profiling people based on their web activities. For example, people are categorized based on the hyperlinks they have followed or based on the content of documents which they have read or downloaded. One of the powerful methodologies to discover groups of related people based on their reviewed documents is latent semantic analysis (LSA) [Berry *et al.*, 1995], [Osinski *et al.*, 2004]. Using LSA technique requires first obtaining a low-dimensional topic-based representation of documents. This is followed by a construction of categories by clustering such representations.

We have extended the concept of LSA to mobile DTNs by representing communities of users' based on documents they have reviewed or locally cached in their mobile devices. In our model, we use singular value decomposition (SVD) as one of the important tools employed by LSA to summarize these documents into low-rank algebraic vectors of interests. These interest-vectors are exploited to discover the underlying latent relationships among users and cluster them into appropriate communities.

Many existing studies have tried to discover social groups of mobile users based on binary relations between users (e.g., the presence of people in the same proximity) [Hui *et al.*, 2007]. Taking a different approach, our model assigns users to appropriate social groups based on their interest-vectors. These interest-vectors, which represent users' interest distributions in different subjects, are extracted from users' locally cached data. Therefore, users with similar interests are implicitly categorized as within the same community of interest, as shown in Figure 4.2. Moreover, because these vectors are extracted from a pool of documents in a user's cache, they can be used to map a new document to a related community of users

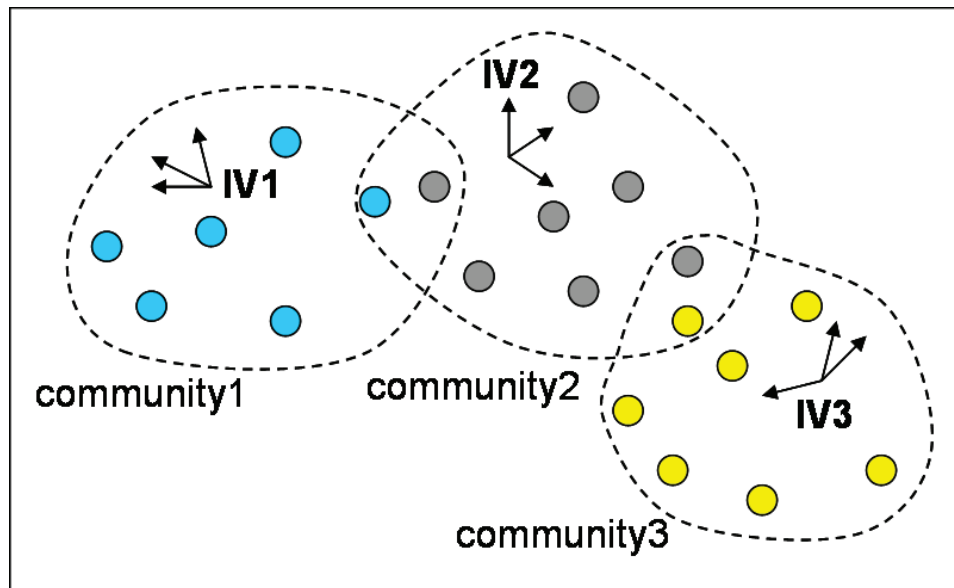


Figure 4.2: **Communities of mobile users.** Mobile users are assigned to appropriate communities of interest based on their interest-vectors (IV). Here, three communities with their respective interest-vectors, IV1, IV2, and IV3, in a multidimensional space, have been displayed.

who have an interest in that document. We have exploited this relationship in our interest-aware routing algorithm to route documents to their interested recipients. Later, in Chapter 5, we use the same concept but a different technique to extract mobile users' interests in different music genres to distribute music files based on users' interests.

4.3.1 Vector space model

As explained earlier, in our model mobile users' interests are extracted from their locally cached data. Therefore the problem of clustering users can be analyzed as the problem of clustering their documents. Consequently, LSA converts the problem of comparing and clustering textual data to a problem of comparing algebraic vectors in a multidimensional space using the vector space model (VSM) [Berry *et al.*, 1995]. In the VSM, every unique term (i.e. word) from the collection of analyzed documents forms a separate dimension and each document is represented by a vector spanning all these dimensions. The relationship between terms and documents is best expressed as term-document matrix. Each element of this matrix is a numerical representation of the frequency of a term in the corresponding document.

4.3.2 Singular value decomposition

LSA attempts to reduce the rank of a term-document matrix in order to eliminate insignificant, noisy words from the data [Berry *et al.*, 1995]. LSA uses an algebraic method of matrix decomposition called singular value decomposition (SVD) to derive the orthogonal basis (singular vectors) of the original term-document matrix. Considering just the most significant singular-vectors, SVD produces a low-dimensional summary of the term-document matrix that exposes the underlying latent concepts of the documents. These top singular-vectors construct the basis of a new, significantly reduced dimensional space that facilitates solving the problem of clustering documents.

SVD singularly breaks down a $t \times d$ matrix A into three matrices, U , Σ , and V , such that $A = U\Sigma V^T$. U is a $t \times t$ orthogonal matrix in which column vectors are called left singular vectors of A , V is a $d \times d$ orthogonal matrix in which the column vectors are called the right singular vectors of A , and Σ is a $t \times d$ diagonal matrix having the singular values

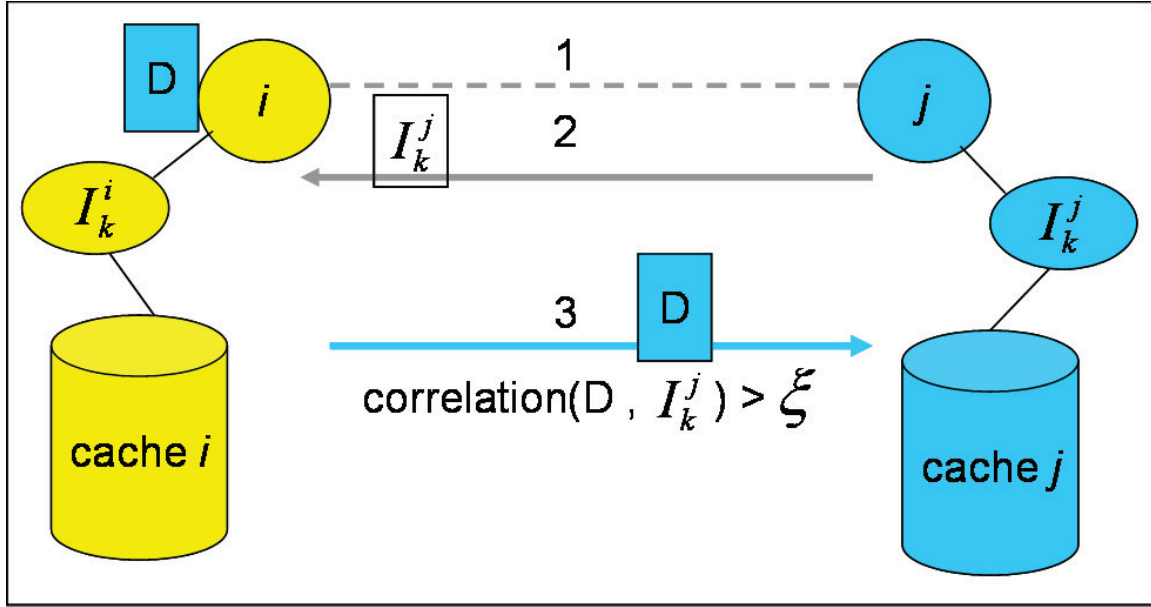


Figure 4.3: **Interest-aware content exchange** 1 - Nodes i , j discover each other. 2 - Node i receives node j 's interest vectors. 3 - Node i relays document D to the node j , if node j 's interests match with the document D .

of A , ordered decreasingly along its diagonal. The rank r_A of matrix A is equal to the number of its non-zero singular values.

These matrices reflect a breakdown of the original relationships into linearly-independent singular-vectors. The use of k largest singular values of A or k -largest singular triplets of $A_k = U_k \Sigma_k V_k^T$ is equivalent to approximating the original term-document matrix A by A_k . The first k columns of U , represented as U_k , form an orthogonal basis for this new approximated space. Orthogonal columns of U_k form the basis of a significantly reduced dimensional feature space or concepts of the original matrix A . In other words, U_k , generated from stored documents in the user's cache, characterizes the user's underlying interests. Therefore, we call the orthogonal vectors of $U_k \Sigma_k^{-1}$ *interest-vectors* and we refer to them as I_k . The right multiplication by diagonal matrix Σ_k^{-1} differentially weighs the separate dimensions.

4.3.3 Interest-aware content distribution protocol

The fundamental concept behind the interest-aware content distribution algorithm is based on matching the documents against the mobile node's interests. In our algorithm, when two mobile users meet, the sender of the document D generates and transfers a copy of the document to the other user, if the document's content matches the other user's interests. In order to determine this correspondence, document D should be mapped into the k -dimensional interest space of the recipient. This mapping is represented as the inner product of the document and the recipient's interest vectors; $\tilde{D} = D^T U_k \Sigma_k^{-1}$, where D is the vector of the words in the document multiplied by the user's interest-vectors. In fact, the elements of vector \tilde{D} are inner products between vector D and the columns of the matrix $U_k \Sigma_k^{-1} \equiv I_k$. Therefore, the normalized elements of the vector \tilde{D} represent the cosine similarities between the original document D and the user's interest-vectors. If there is enough correlation between the document's latent concepts and the user's interest space, the document D is transferred to the other mobile user. The steps of the interest-aware content exchange algorithm are shown in the Figure 4.3. Node i is represented as the sender of the document D . After meeting node j , node j 's interest-vectors are transferred to node i . A copy of document D is generated and transferred to the node j if the correlation between this document and node j 's interest space is greater than some threshold value, ξ . The threshold value ξ is determined based on the application's policies.

4.4 Simulation setup

We have evaluated the performance of our interest-aware content distribution protocol by simulations. Simulation analysis of the routing protocols for mobile networks depends heavily on the characteristics of the forwarding algorithms as well as of the mobility of mobile users. The most important factor in evaluating routing protocols by simulation is how well the assumptions about the forwarding algorithm and synthesized mobility traces reflect the real-world traits. In this section we explain our interest-aware protocol implementation in the ONE [ONE, 2007] simulator and our choice of mobility traces for its performance evaluation.

ONE is a simulator specifically designed for mobile DTN scenarios and is capable of interfacing with real-world as well as synthetic mobility traces. We have imported mobile phone traces from the reality-mining database [Mining, 2003] into the ONE simulator to achieve a more realistic evaluation of our interest-aware content distribution algorithm. The reality-mining database is one of the largest mobile phone data collected in academia [Eagle and Pentland, 2006a].

4.4.1 Implementation of the interest-aware protocol in the ONE simulator

The ONE simulator is a discrete event simulation environment that can be expanded by adding new modules to its routing and event generator packages [Kernen and Ott, 2007]. We have added our interest-aware forwarding algorithm as a new module to the ONE's routing package. The original implementation of the ONE simulator was based on a single-source, single-destination communication model, and there was no notion of document categories. Therefore, we have extended the ONE simulator's source code to implement group-based communication and to create data content categories.

During each simulation round, we randomly generate a sub-population of mobile users as the target audience for some specific data content. We adjust this sub-population's interest-vectors to distinguish them as the target recipient's of that specific data. We also generate a finite number of communities, each of which is interested in some category of data content. The rest of the population of mobile users is assigned some random interest vectors. In our simulation, at each simulation round, some mobile nodes are selected arbitrarily as the senders of some data document. The document content is synthesized in the sender in order to meet one of the communities' interests. One restriction that we apply to choose the sender of a specific document is to select it from the entire population, excluding the target recipients of that document. In our simulation scenarios, the document size is fixed and every mobile user has a limited cache size which is set to hold a maximum number of 5 million documents each with size of 1KB. This totals 5GB, which is the usual storage size for a handheld device at the time of this research.

The implementation of the forwarding algorithm is based on the interest-aware com-

munication model described in Section 4.3. When two mobile users meet and one of them has some document objects in its cache, those documents are transferred to the other user if, and only if, the documents content is correlated with the other user's interest-vectors. In other words, in our algorithm documents are transmitted toward the target community of those documents through intermediate mobile users or relays who are also interested in that document's content.

The simulation evaluates the performance of the interest-aware forwarding algorithm in distributing as many relevant documents as possible among the users who are interested in those documents' content. In other words, we evaluate the coverage of communities of interests with their content of interest. We compare the performance of our algorithm with the epidemic distribution of data content. We have chosen epidemic routing as our comparison for two reasons: first, because, assuming unlimited communication resources such as storage and bandwidth, epidemic routing has been proven to produce the fastest data distribution. Second, because all other important routing algorithms which are proposed for mobile DTNs do not address data transmission to a group of recipients. In our simulations, the mobility of mobile users is based on the reality-mining traces [Mining, 2003].

4.4.2 Reality-mining data traces

Previous studies show that the performance of mobile DTN routing algorithms depends heavily on the choice of the mobility model [Abdulla and Simon, 2007], [Chaintreau *et al.*, 2007]. DTN forwarding algorithms demonstrate different performance results using synthetic or real-world mobility traces. Therefore, in order to have a more realistic evaluation of our interest-aware routing algorithm, we have decided to use the reality-mining database [Eagle and Pentland, 2006a].

In the reality-mining study, almost 100 Nokia model 6600 smart phones were pre-installed with several pieces of software developed by the Massachusetts Institute of Technology as well as with a version of the Context application from the University of Helsinki [Raento *et al.*, 2005]. The dataset, which is a MySQL relational database, was collected for the purpose of monitoring mobile phone usage behavior in order to model complex social systems. Among different information collected using the mobile phones, we used the Blue-

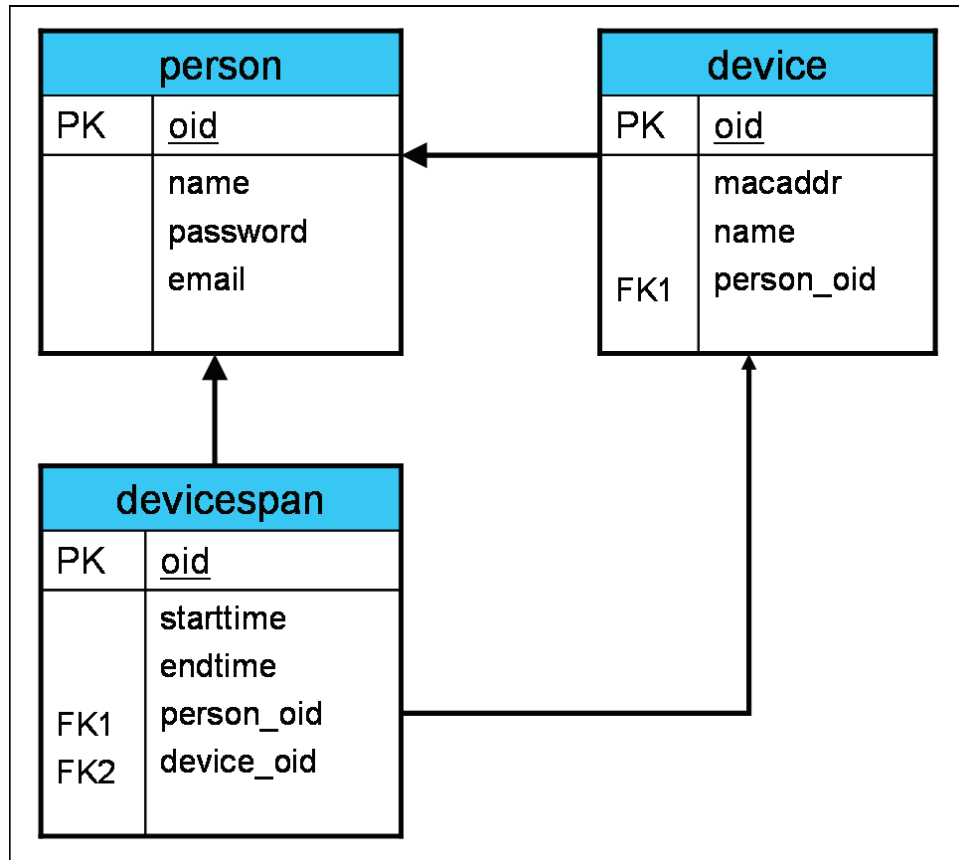


Figure 4.4: Bluetooth device-to-device encounter tables as part of the reality-mining relational database. The MySQL table, represented as *person*, records the information about the owner of the device. The *devicespan* table contains encounters with other bluetooth devices either part of the reality-mining project or devices which belong to mobile users outside this experiment.

tooth device-to-device encounter logs that have been represented as MySQL relational tables represented in Figure 4.4. Based on our extracted statistics from the reality-mining relational database, the total number of Bluetooth devices recorded in this study is 20,795. Out of these 20,795 recorded Bluetooth devices, 103 of them are Nokia model 6600 mobile phones that belong to the participants in the reality-mining experiments. Therefore, we extracted two different types of contact events from these traces. The "participant-participant" contact events comprise the contact events and their durations that occurred only between the reality-mining participants. On the other hand, the "Bluetooth-Bluetooth" contact events contain both the encounters between the reality-mining participants as well as contact events between the devices of participants' and non-participants. Then, we converted these two extracted traces to the format accepted by the ONE simulator's event generator module. The consideration of both these traces in evaluating our interest-aware routing algorithm enables us to study the contribution of data exchanges with Bluetooth devices that are not directly participating in the reality-mining study in the dissemination of information among the reality-mining community.

4.5 Evaluation of the interest-aware content distribution protocol

We simulated the interest-aware content distribution and evaluate its performance in this section. In our simulations, mobile nodes and their contact events are generated based on the reality mining mobility traces. As we explained in the previous section, these traces record the contacts among users participating in reality-mining as well as contacts between reality-mining participant and non-participant Bluetooth devices. In our simulations we are specifically interested in data exchanges among mobile users participating in the reality mining project or in other words our participant-participant contact traces. We have also studied the role of the encounters between participant and non-participant devices in data distribution by using our extracted Bluetooth-Bluetooth traces.

In our simulations, we randomly select a community of interest as a fixed percentage of the total reality-mining participants, and we assign them similar interest-vectors. We

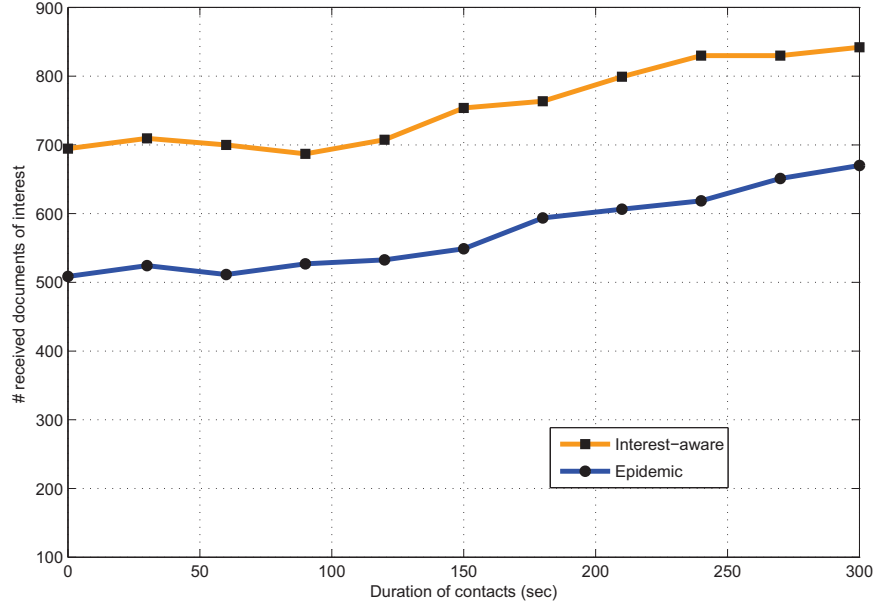


Figure 4.5: Coverage of the community of interest.

changed this size to vary from 10% to 60% of the total number of the participants. Because we did not see any significant difference in the final evaluations, the graphs represented here are related to the average community size of 35% of the total participants. This subpopulation of mobile users is implicitly assigned to the same community of interest, and in our simulations it is considered as the target recipients. Furthermore, in the simulations we have specified a fixed number of categories for the documents and all other mobile users are randomly assigned to have some interest in each of these categories, uniformly ranging from 0 to 10%. In reality, the number of categories of interests highly depends on the personal attributes of the mobile users. Although our simulation results have been presented for a moderate number of 15 general categories, this number is not a defining factor in the performance of our algorithm. All documents are generated from some sources which are randomly selected from the remaining 65% of the mobile users who do not belong to the community of target recipients. We specifically track the distribution of those documents with content closer to the interests of our pre-specified target community. The similarity of the data content to the interests of the mobile users is calculated in the vector space model

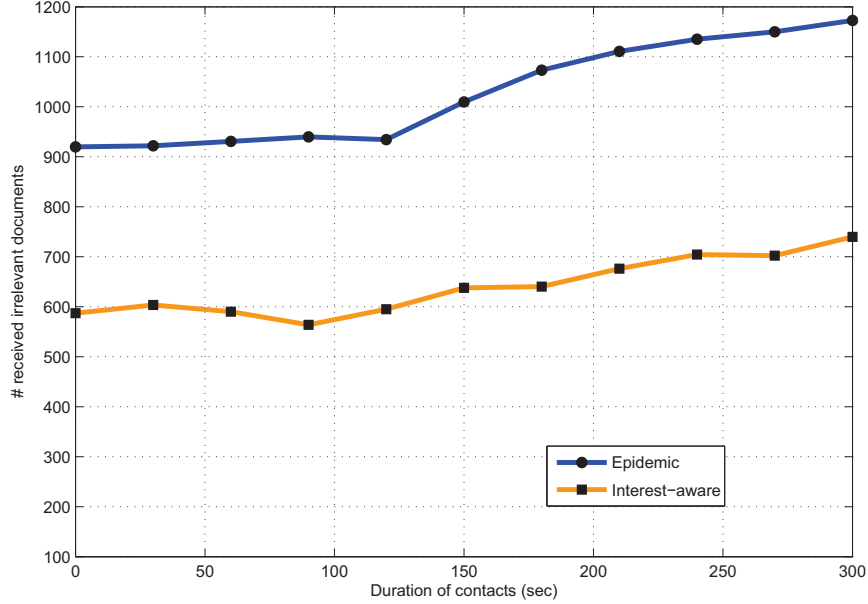


Figure 4.6: Distribution of irrelevant documents among mobile users.

which is derived from the LSA technique as described in 4.3.1. We choose the similarity threshold of the algorithm to be 50% in our simulations.

Contact traces of the reality-mining database have been recorded with a temporal resolution of 300 seconds. The analysis of the reality-mining traces reveals that almost 44% of the observed contacts that might have lasted less than 300 seconds, are recorded with a duration of zero. So, in running the simulations for the original reality-mining data traces, there will not be any data exchange in 44% of the contact opportunities. Therefore, as well as running our simulations for the original traces, we have synthetically increased each contact duration by 30 seconds at each round. This can be interpreted as increasing the co-residence time of mobile nodes in the same proximity which is the case for networks with less mobility or mobile devices with longer radio ranges than those of Bluetooth.

4.5.1 Evaluation of the protocol using participant-participant traces

For the series of the simulations described in this section, we used participant-participant contact records between the reality-mining participants. Figure 4.5 shows the number of

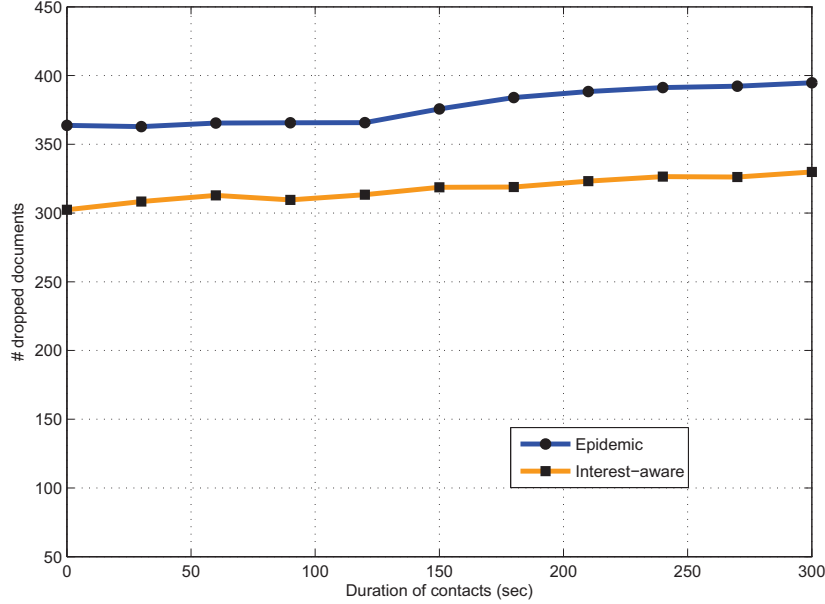


Figure 4.7: Total number of dropped documents in mobile devices due to cache overflow.

documents of interest that are received by the mobile users in the target community after each simulation run. The abscissa represents contact durations that are increased in each run from zero to 300 seconds. To decrease the impact of the random document generation in the final simulation results, each data point on the graph has been calculated as the mean value of the 20 trials of the same simulation. As we see in the graph, the coverage of target recipients increases as the duration of contacts increases. This is the natural outcome of providing more data exchange opportunities for mobile users. As we can see, however, our interest-aware content distribution protocol maintains its superiority over the epidemic routing for the average of 30% all over the simulations. Interest-aware keeps this superiority at the same time it avoids flooding mobile users with irrelevant data content. This fact can be verified by referring to Figure 4.6. This figure shows that epidemic routing distributes a higher number of irrelevant documents, with respect to mobile users' interests when compared to our interest-aware protocol. On average, our interest-aware protocol distributes 35% fewer irrelevant documents.

This improved performance occurs because our algorithm limits the data exchange to

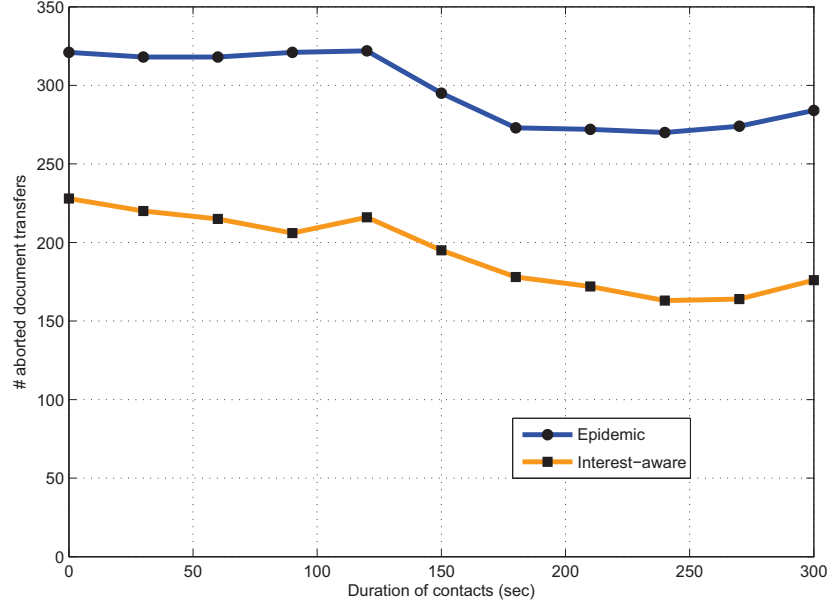


Figure 4.8: Total aborted transfers due to insufficient contact times.

only the documents relevant to the interests of the mobile users. Therefore, the interest-aware algorithm has a more efficient usage of communication resources in terms of cache and wireless inter-contact times. In our simulations, mobile devices have a limited cache size of 5 million documents. Documents that are going to be transferred or documents that are received from the other mobile nodes are stored in the cache. When the cache is full, our algorithm removes the older documents from the cache. By preventing the distribution of irrelevant data, our algorithm preserves cache usage mostly for the relevant documents. Consequently, documents intended for the target recipients are saved from being tossed out in intermediate nodes due to the cache overflow more often than the epidemic approach allows. Figure 4.7 shows that using our protocol in our simulations reduces the number of dropped documents in a mobile devices' cache by an average of 16% as compared to epidemic distribution.

In reality, inter-contact times between mobile users are limited. Therefore, depending on the amount of data that is exchanged at each contact opportunity, the data transfer might be terminated before completion. By using the interest-aware algorithm, which limits the

document exchange at each wireless contact to only the relevant documents, the number of unfinished data transfers decreases significantly. Figure 4.8 shows that in our simulations, the number of unnecessary exchanges at each wireless contact decreases on average by 35%, compared to epidemic routing. Therefore, by using our interest-aware algorithm the number of the relevant documents that reach their target recipients increases.

4.5.2 Evaluation of the protocol using Bluetooth-Bluetooth traces

In these series of simulations we used the contact events that occurred between reality-mining participants and non-participant Bluetooth devices. We initially expected to achieve a better performance result in terms of the coverage because contact opportunities between participant and non-participant devices should potentially enhance the document distribution and subsequently increase the coverage of the target recipients. Moreover, while the main purpose of the reality-mining trace logs is to capture encounters among participants themselves, in the reality-mining study the majority of the encounters recorded in these traces are between participant and non-participant devices (171,466 out of 285,512 total encounters). However, after investigating the performance results, we did not observe much improvement in terms of the coverage. By collecting more statistics from the reality-mining's original database, we realized that, out of the total number of 20,692 non-participant devices, 14,029 were seen just once in the entire trace log. In other words, 68% of the non-participant devices never were encountered again by any of the reality-mining participants. Therefore, these external contacts contribute very little to the data distribution among reality-mining participants.

4.6 Conclusion

Opportunistic routing in mobile DTNs intrinsically exploits nodes movement and human social interactions to overcome network partitions. In the research community it is widely believed that identifying community information about recipients can select suitable forwarders and reduce the delivery cost compared to the naive "oblivious" flooding [Hui *et al.*, 2007]. The reason for this assumption is that people in the same community are likely

to meet more regularly and hence be appropriate forwarders of messages destined for other members of their community. Therefore, the knowledge of users' social communities and their interests helps in the design of practical applications and routing algorithms for mobile DTNs. In the research, so far, social interactions of mobile users have been specifically associated with their geographical locations and their physical device-to-device contacts. In reality, of course, social relations among mobile users are much more complex than the fact that mobile users just happen to be in the vicinity of other users. Furthermore, the history of contacts is not the only indication of social connections. For example, two people who meet for the first time and do not share any contact history might actually have some common interests and, therefore, be able to provide each other with useful information. Based on the current routing algorithms, however, interest- matching between these two users is neglected for data-routing purposes [Miklas *et al.*, 2007], [Daly and Haahr, 2007].

Interest-aware content distribution protocol for mobile DTNs provides a new tool to measure the social relationships between mobile users based on their interests. With the use of such an algorithm, the interaction between mobile users is based on the correlation in their interests. The interest-aware algorithm is capable of inferring mobile users' interests from their cached content and the history of the documents they have reviewed. Our algorithm uses this inferred information to provide mobile users with data content that matches their interests. Our interest-aware content distribution algorithm uses the vector-space model (VSM) as part of LSA technique to represent documents as algebraic vectors in real coordinate-space. It then uses singular value decomposition (SVD) to produce a low-dimensional summary of the documents stored in each user's cache. In addition, LSA uses this low-dimensional representation to cluster documents into appropriate categories which characterizes a user's interests. Users' interest categories are represented by their interest-vectors, which are extracted from the pool of documents cached in the users' devices. When two mobile nodes meet, these interest-vectors are exchanged as an indication of the user's interests in the documents the other node has to offer.

We implemented our interest-aware protocol in the ONE simulator and ran our simulations using the reality-mining mobile phone traces. Our simulation results show that, compared to the epidemic routing with the same storage and bandwidth constraints, our

protocol has a higher coverage of mobile users with the content of interest to them. Furthermore, our interest-aware algorithm distributes much less irrelevant data among mobile users. In our simulation scenario the coverage of mobile users with our algorithm is 30% higher than epidemic and results in 35% lower distribution of irrelevant data. Furthermore, the number of aborted data transfers due to insufficient wireless contact duration decreases by 35% using our interest-aware algorithm. Therefore, the knowledge of interests provides a more efficient routing mechanism and saves for communication resources such as disk space and wireless contact opportunities.

Chapter 5

PEEP: Popularity Based and Energy Efficient Content Distribution Protocol for Mobile DTNs

As we explained in the previous chapters, all routing algorithms for content distribution in DTNs communicate based on a store-carry-forward paradigm. Store-carry-forward communication withstands the intermittent connectivity caused by mobility and low node density by exploiting a mobility-assisted routing strategy: nodes meet, receive data, hold the data in storage, and wait for new contact opportunities to transfer that data to other mobile nodes. In Section 4.2, we broadly reviewed the previously introduced routing protocols for mobile DTNs, and we concluded that the main focus of these routing protocols has been on the single-source, single-destination communication model and thus the main challenges of a practical multicast architecture still remain unaddressed. Despite the challenging nature of multicast for mobile DTNs, this communication model is necessary when data needs to be transmitted to groups of recipients. Proposed models for multicasting in mobile DTNs try to extend the classical multicast model to mobile DTNs [Zhao *et al.*, 2005],[Chen *et al.*, 2006]. This is not feasible for these types of networks, which are characterized by a

frequently changing topology and lack of infrastructure. The only feasible multicast models for mobile DTNs are epidemic multicast [Vahdat and Becker, 2000] and the interest-aware content distribution protocol which was introduced in Section 4.3.

Epidemic multicast doesn't require any previous knowledge about network topology and is proven to provide an upper bound in information propagation speed in mobile DTNs [Jacquet *et al.*, 2010]. This best possible upper bound, however, has been calculated without consideration of the inadequacies of epidemic routing. The main problem with the epidemic routing is inefficient usage of communication resources such as disk space, transmission energy, and bandwidth. The performance of epidemic routing significantly degrades when data transmission is aborted prematurely due to short wireless contact opportunities or insufficient battery life. In addition, the poor performance of epidemic propagation results from the cache overflow caused by exchanging large amounts of data while having limited disk space. While storage space overflow may become insignificant in the future, due to the dramatic increase in mobile storage space, energy is still a restrictive factor.

Today, mobile devices are equipped with variety of functionality such as wireless interfaces, GPS, and multimedia. All these mobile technologies consume energy and place growing pressure on devices' battery life. Furthermore, advances in battery design, especially improvements in lithium-ion battery capacity, have stalled for the past few years, and the growth of power-saving technologies is currently not as fast as the increases in energy consumption by mobile devices. Considering this energy limitation, the deployment success of new mobile technologies, including mobile DTNs, depends highly on their battery consumption.

Another important factor in the wide acceptance and deployment of mobile DTN technologies is user's personal benefit from employing these technologies. Routing algorithms that are designed for mobile DTNs mostly transmit data to its destination through intermediate nodes, also known as relays, which might not be interested in the data themselves. This results in mobile users utilizing their communication resources to carry data without any personal gain.

Interest-aware data routing algorithms, as explained in detail in Chapter 4, consider mobile users' personal interests in their routing decisions. These algorithms help the par-

participating users in the routing to locate and extract the data content that interests them. This gives mobile users a stronger incentive to share their communication resources to carry other users' data, as they also benefit from it.

A major drawback of the interest-aware algorithm is its greediness, meaning it dedicates each wireless contact opportunity to transmit only data items which are of the interest to the mobile users present at the time of that contact. By just focusing on the interests of the mobile users involved in the current wireless contact, hence this algorithm does not help the other mobile users who are few more hops away to receive their items of interest. Therefore, interest-aware algorithms might eventually cause the entire mobile DTN to starve. Another important deployment limitation of the interest-aware protocol is its lack of energy awareness which, as explained before, is another main limiting factor that mitigates against deployment of mobile technologies.

In this chapter we evolve the interest-aware communication model toward a more comprehensive model which we name PEEP. PEEP solves interest-aware greediness and energy consumption issues. We solve the energy issue by introducing transmit budgets. The transmit budget is a data transmission token which determines the amount of data allowed to be transferred over each wireless contact. The transmit budget is based on the policies enforced by the user or the application. We also propose different transmit budget allocation methods in the PEEP protocol to avoid the greedy behavior of the interest-aware communication model. The main idea behind these allocation methods is to divide the transmit budget into two portions. The first portion is dedicated to transfer items of interest for the mobile users who are involved in the current wireless contact: the second is allocated to transfer "other" data items which are estimated to be useful for the mobile users who will be encountered in the future. The details of the transmit budget allocation methods are explained in Section 5.2.

We implement the PEEP routing protocol in the ONE simulator and evaluate its performance in distributing items of interest. We show that dedicating some portion of the transmit budget to transfer other items that might be useful for the future contacts significantly improves the performance compared to the previous multicast models, both epidemic and interest-aware. Our simulations show that the PEEP protocol distributes items of in-

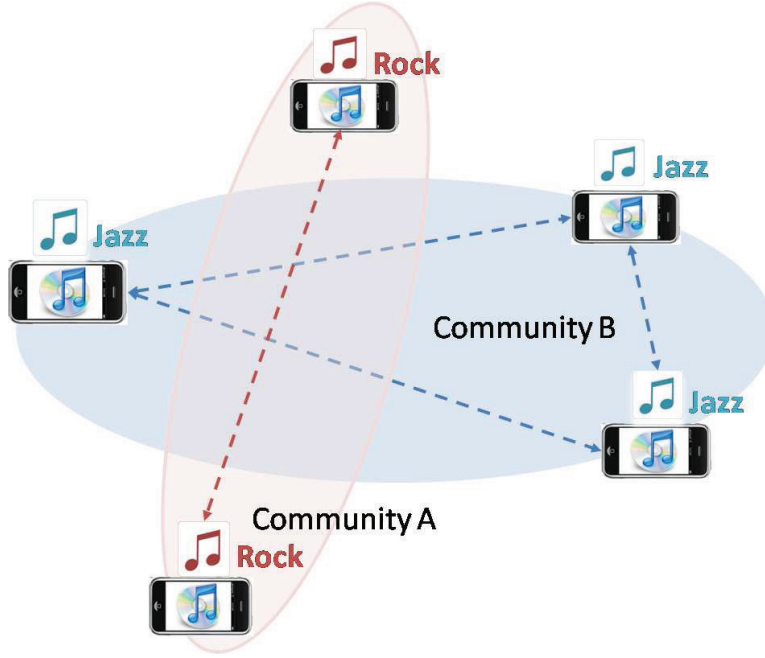


Figure 5.1: **Example of different communities of interests.** Our interest-aware music sharing application automatically shares music among users with similar taste in music.

terest among mobile users, on average, 44% faster than the legacy multicast mechanisms. We further prove that PEEP’s performance is robust and does not depend on the underlying mobility model of the mobile users.

In Section 5.1 we motivate the need for an interest-aware group communication model for mobile DTNs. Section 5.2 elaborates on our PEEP routing algorithm for mobile DTNs. In Section 5.3 we explain our implementation scenario and evaluate the simulation results. Finally, Section 5.4 concludes this chapter.

5.1 Feasibility of an interest-aware architecture for mobile DTNs

With the increasing amount of storage space and different networking capabilities of today’s mobile devices, users often store a large quantity of content such as music, videos, or news articles on their devices. We argue that exploiting this data to extract a meaningful representation of an individual user’s interests could be the foundation of the next generation

of intelligent multicast service providing in mobile DTNs.

The problem of multicasting to groups of users has been studied in mobile DTNs [Zhao *et al.*, 2005]. This proposed multicast architecture is based on a group membership model, in which mobile users register with a specific group to exchange multicast messages with other group members. However, due to the lack of any knowledge about network topology and any infrastructure to manage group memberships, implementation of this multicast model for mobile DTNs is infeasible.

In Chapter 4 we introduced interest-aware algorithms as a new group communication paradigm to solve multicast issues in mobile DTNs [Arezu Moghadam and Henning Schulzrinne, 2009]. Interest-aware communication algorithms assist mobile users to locate and receive their content of interest automatically. For example, using this algorithm, when mobile nodes are in physical proximity, they can automatically share movie or restaurant reviews as well as music files with other users who have similar tastes. By using epidemic multicast, however, every mobile user present in the wireless contact is flooded with all reviews or music files of the others. As a proof of concept, we have implemented an interest-aware music and news sharing application for mobile DTNs that automatically learns users' interests from their past behavior. Using the information about a user's interests, our application discovers and downloads music files and news articles into the user's device from other mobile users who have similar tastes. Our interest-aware application discovers and extracts the user's favorite files automatically without human intervention.

5.1.1 Interest-aware music and news sharing application

An example scenario of the interest-aware music sharing is presented in Figure 5.1. In this scenario, there are two groups of mobile users: community A is interested in rock music, while community B is interested in jazz. Our interest-aware, music sharing application learns about users' individual interests and automatically assigns them to the appropriate related communities. When all these mobile users are in physical proximity, their devices discover other users' presence. Then, instead of flooding all users, our interest-aware music sharing application automatically downloads music only from users with the same taste in music.

In this application the knowledge of the interests of mobile users is learned based on their past behavior. Each user's interest in a specific genre is extracted based on the number of times the user listens to a specific type of music and the dates and times the user plays that music. Based on our implementation, the most recently and most frequently played songs indicate that user's interest in that specific genre. We have implemented this application on top of our 7DS BonAHA platform, using its APIs for device and service discovery [Moghadam *et al.*, 2006].

5.1.2 Popularity

In this chapter we evolve the interest-aware communication model to a more comprehensive algorithm, PEEP. In addition to the individual interests of mobile users, this algorithm estimates the global popularity of the data items and saves transmission energy by defining a *transmit budget*. Transmit budget is a token that limits the number of data items that can be transmitted during each wireless contact. By limiting the data transmission over each contact, PEEP exploits the contact opportunity in a more efficient way by transmitting the most useful data. Therefore, PEEP is able to avoid the premature termination of data transmission, especially when duration of wireless contacts is short. This is particularly important in highly mobile scenarios which are characterized by short contact durations. Because the amount of traffic over each wireless contact is regulated by a transmit budget, an intelligent algorithm is required to select the best data items to be transferred over the contact. The best data items are the ones that are either of interest to the first-hop recipient or considered globally popular and thus of interest to users who are encountered in the future; for example, market news might be popular in the Wall Street neighborhood. The PEEP algorithm estimates the global popularity of data items beyond the immediate first-hop wireless contacts. PEEP estimates global popularity by recording the history of the interests of the recently-encountered mobile users. PEEP allocates some portion of the transmit budget to transmit globally popular data.

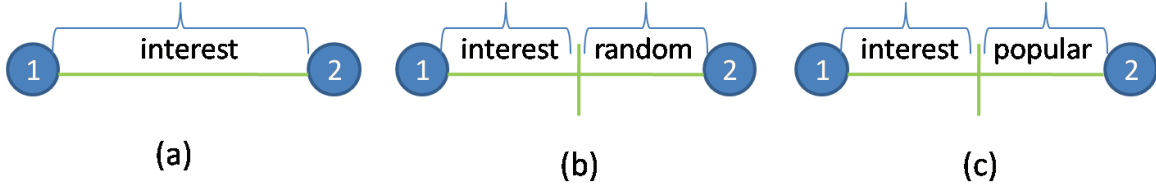


Figure 5.2: **Budget allocation methods in the PEEP algorithm.** a) Interest-Only in which only items of interest are transmitted over the contact. b) Interest-Rand in which transmit budget is divided between items of interest and other randomly selected data items. c) Interest-EstPop and Interest-GlobPop in which the second portion of the transmit budget is assigned to transmit popular items.

5.2 PEEP distribution algorithm

In this section we elaborate on the PEEP algorithm and its mechanism for learning the personal interests of mobile users and the global popularities of data items. We also explain different possible measures PEEP takes into account in order to partition the transmit budget into items of personal interest and items that are globally popular across the network. The first step in distributing content of interest among mobile users is to extract users' personal interest in the available data content. Every media type such as pictures, music, movies, and documents can be categorized into genres based on the type of its content. Some genres might be more popular among users based on age, gender, profession, and cultural or political viewpoints. Each genre, therefore, has a certain type of audience or, in other words, people have different interests in different categories. Different methods and algorithms have been introduced to recognize users' interests in order to provide them with relevant content. The most obvious method to identify users' interests is to get users' direct feedback, for example, via survey forms. In the context of web usage mining [Grcar *et al.*, 2006], personalized information delivery is achieved by profiling Internet users based on the hyperlinks they follow or the content of web objects they download. In mobile DTNs, users are generally profiled based on their mobility behavior [Hsu *et al.*, 2008], [Leguay *et al.*, 2005], [Boldrini *et al.*, 2007]. Interest-aware algorithms, on the other hand, propose a different classification method to categorize users in communities of interest. They do so based on the type of web documents which users store in their devices. We

explained the details of interest-aware algorithms in Chapter 4. Here, we assume that the number of categories of interest is finite and that the appropriate tools to extract users' interests in these categories are available, using any of the previously discussed methods. After extracting users' interest, these interests are represented in a high-level format called *interest vectors*.

5.2.1 Interest vectors

Here, we simplify the definition of the interest vectors slightly from what we defined in Chapter 4. We assume C to be the predefined and finite number of categories or genres. The interest vector of mobile node n , represented as I_n , is a binary vector or bit array of size C . These binary representation of the interest-vectors is the simplified version of the general definition we provided in Chapter 4. Further, elements of this binary vector, I_n , express a user's interest in each category or genre with 0 indicating no interest and 1 indicating some interest.

5.2.2 Transmit budget

The transmit budget of mobile device, n , represented as K_n , indicates the maximum number of data items this device is allowed to transmit during each wireless contact. By limiting the data transmission, the transmit budget provides an energy saving tool for mobile devices compared to the other routing algorithms designed for mobile DTNs. Since the amount of the transmit budget is determined by mobile users and applications, the transmit budget offers a higher degree of freedom to enforce energy management policies.

5.2.3 PEEP algorithm

The PEEP algorithm proposes different methods to partition the transmit budget to transfer items of interest and other items. The goal of the transmit budget allocation is to maximize distribution of items of interest while avoiding the greedy behavior of the interest-aware communication model [Arezu Moghadam and Henning Schulzrinne, 2009]. Another important objective of the PEEP algorithm is to minimize the delay in distributing items of interest throughout the network. The main constraint of the PEEP algorithm is the

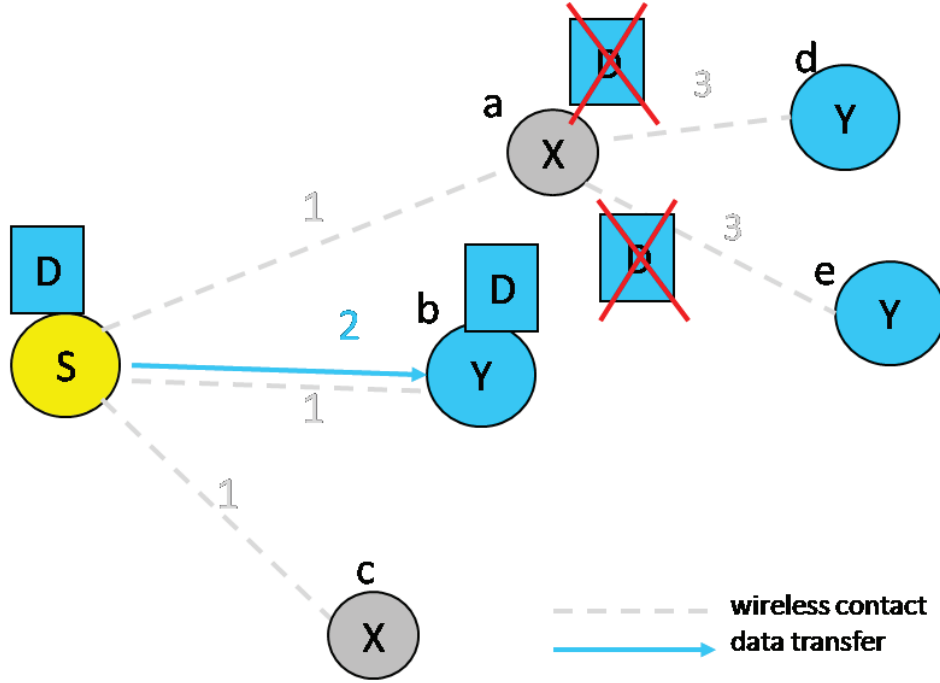


Figure 5.3: **Greedy behavior of the Interest-Only model.** 1. The source (S) estimates the potential recipients of data content (D) as belonging to community Y . 2. After calculating the similarity between the interests of the encountered users and data content D , Interest-only model transfers the data to the most similar user to group Y , which is node b . 3. Node b meets new mobile users. 4. After similarity calculations, two mobile nodes are selected as the interested recipients of the data content, nodes e and g .

transmit budget, which limits data transmission over each contact. In other words, the objectives of the algorithm could be rephrased as maximizing the distribution of items of interest, minimizing the delay of the distribution, and minimizing the amount of energy consumed for this distribution. Here, we formulate our optimization problem considering the two first objectives; the number of useful items distributed by the PEEP algorithm and the delay of this distribution. However, these two objectives are not as independent as they might seem to be. If we assume the goal of the algorithm is to distribute a total number of X useful items among all mobile users, all propagation algorithms as well as the epidemic will eventually achieve this goal. Their main difference would be the time they require to propagate this number of items. Therefore, we summarize our optimization criterion to

the minimization of the delay. Based on this discussion, the problem can be formulated as follows: Suppose that N_i mobile users are interested in category C_i and that S_i is the set of nodes interested in category C_i . Based on this definition:

$$|S_i| = N_i$$

Let t_{ij} be the time it takes for data item of category i to reach node j . Therefore, the objective function of our optimization, constrained to the transmit budget, is

$$\min \frac{\sum_i \sum_{j \in S_j} t_{ij}}{\sum_i N_i}$$

Different methods of dividing the transmit budget between items of interest and items of other categories are represented in Figure 5.2. These mechanisms are explained in the following sections.

5.2.3.1 Only items of interest(Interest-Only)

The simplest model, as shown in Figure 5.2 (a), is to assign the entire transmit budget to just the items of interest. This model is actually similar to the interest-aware communication model with a transmit budget constraint. The drawbacks of this method are the following: First, if the number of items of interest is less than the transmit budget, the entire transmit budget or some portion of it is wasted. Second, this method leads to greedy behavior and might cause the entire DTN to starve. The greedy behavior of the Interest-Only method is shown in Figure 5.3. In this scenario, nodes d and e which are interested in data item D are not able to receive it, because they are one hop further from node a which is not interested in receiving data item D .

5.2.3.2 Items of interest plus random items (Interest-Rand)

In this method of the transmit budget allocation, if there is extra budget left after two nodes transmit all items of interest, the nodes fill the remainder of the transmit budget to send other items which are randomly selected from different categories. This communication

model is represented in Figure 5.2 (b). The advantage of this technique is more efficient usage of the transmit budget and avoidance of the greedy behavior of the Interest-Only mechanism. Using this model, items of other categories as well as data items from categories of interest find a chance to diffuse into the network.

5.2.3.3 Items of interest plus popular items

Unlike the interest-aware algorithm, PEEP differentiates between the concept of individual interest and popularity. Here, an individual interest is the personal interest in each category of a specific mobile user, while popularity represents a more global perception of interests throughout the network. By this definition a user might not be personally interested in a popular item. For example, a mobile user who is visiting the Columbia University campus might not be interested in receiving any campus-related news, while this news subject is of the interest of the majority of people on the campus and, therefore, popular. Considering this fact, scheduling popular items for transmission over a wireless contact instead of the random selection, as in the Interest-Rand model, provides a better opportunity for mobile users to receive data items of their interest. This communication method, as represented in Figure 5.2 (c), schedules the globally popular items for transmission in the second portion of the transmit budget. This allocation method of the PEEP algorithm, however, requires knowledge of global popularity for a successful performance. In an ideal case the knowledge about the global popularity of data items is provided through some external source or "oracle". We refer to this model of the PEEP algorithm as Interest-GlobPop. Although optimal, the assumption about the existence of an oracle is impractical, at least in mobile DTNs, due to the lack of any global information about nodes' identities and interests. The PEEP algorithm solves this issue by learning and estimating popularity based on the information derived from the history of past contacts.

5.2.3.4 Popularity estimation (Interest-EstPop)

Having exact knowledge of the global popularity of genres is almost impossible in an infrastructure-less mobile environment, where mobile users frequently join and leave the network. A practical mechanism for mobile users to learn about global popularity is to

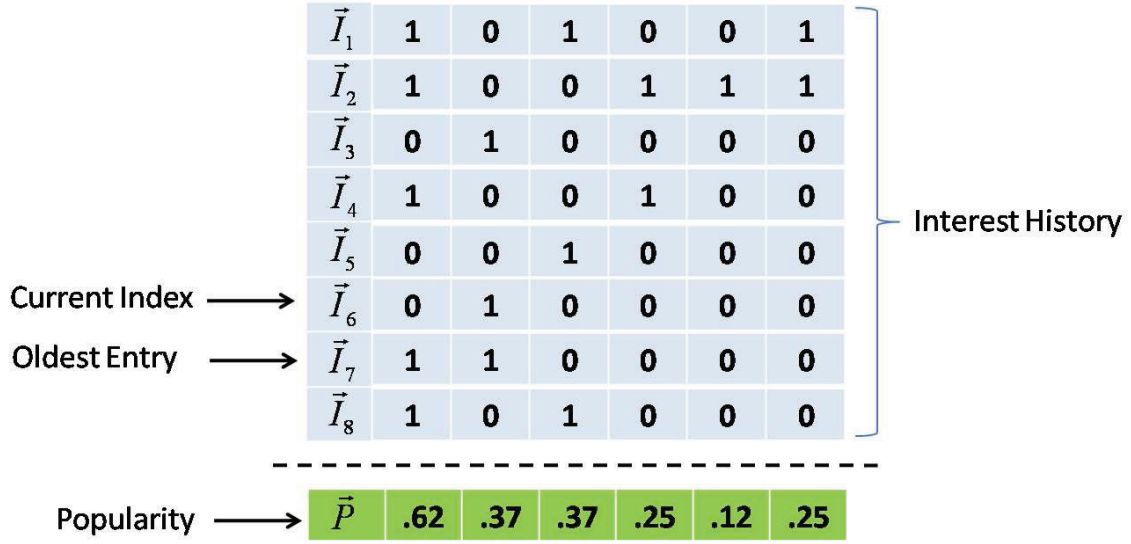


Figure 5.4: **Example of interest history window.** Current index points to the most recent entry in the interest history buffer. Here, popularity is calculated based on the popularity estimation formula as the average of the interest vector entries in the buffer with $\alpha_i = 1$.

monitor the interests of the users which they have encountered in the past. Popularity estimation by the PEEP algorithm is based on learning the global behavior by monitoring local patterns. Each mobile user stores interest vectors of the few most recent contacts in a buffer called *interest history window*. The accuracy of the popularity estimation at a node depends on the size of the interest history window at this node. In the PEEP algorithm, the global popularity of the categories is calculated as the weighted average over the interest vectors stored in the interest history window with size W as follows.

$$\vec{P} = \frac{1}{N} \sum_{i=1}^W \alpha_i \vec{I}_i$$

$$\forall i \alpha_i > 0$$

\vec{P} is a vector of real numbers whose elements represent the values of the estimation of the popularity for each category or genre, and α_i weighs the importance of each contact at each point of history in the interest history window. An example of interest history window is shown in Figure 5.4. The PEEP algorithm avoids the overflow of the interest history

buffer by replacing the oldest stored interest vector entry with the newest one.

5.3 Simulation results

We evaluate the performance of the PEEP content distribution protocol by simulations. Since the delay and the rate of the data distribution have an inverse relationship, we use the rate of the distribution of items of interest as our comparison factor among different methods of the PEEP algorithm in partitioning the transmit budget. Simulation analysis of the routing protocols for mobile networks depends heavily on the characteristics of the forwarding algorithms as well as mobility of mobile users. In mobile DTNs the choice of the mobility model governs the number and duration of the wireless contacts, which in turn impacts the speed of the data distribution. Here, we discuss that in order to have a robust evaluation, the final performance of the PEEP content distribution protocol should be independent of the choice of the mobility model. Below, we explain the PEEP protocol's implementation in the ONE simulator [ONE, 2007], and we show that the choice of mobility model does not affect PEEP's performance results. The ONE is a simulator specifically designed for mobile DTN scenarios and is capable of interfacing with real-world as well as synthetic mobility and message generation traces.

5.3.1 PEEP's implementation in the ONE simulator

The ONE simulator is a discrete event simulation environment. Its source code is in Java and can be expanded by extending its routing and event generator classes [Kernen and Ott, 2007]. We have added the Interest-Only, Interest-Rand, Interest-Pop, and Interest-GlobPop budget allocation models of the PEEP forwarding algorithm as new modules in the ONE's routing package. The original implementation of the ONE simulator was based on a one-to-one communication paradigm, and there was no notion of data categories. Therefore, we have extended the ONE simulator source code to implement group-based communication and to create data content categories.

In our implementation, the number of the categories of data content and therefore the size of the interest vectors are finite and fixed. In reality, the number of categories of interests

depends highly on the mobile users' demography. Although our simulation results have been represented for a moderate number of 15 general categories, this number is not a defining factor in the performance of our algorithm. The global popularity of the categories in our simulation scenario is assigned based on Zipf distribution. This popularity assignment originates from the previously studied and widely used phenomenon in the web content caching in the Internet [Adamic and Huberman, 2002]. Based on users' Internet usage traits, millions of users flock to a few selected sites, giving little attention to millions of others. In a similar fashion many mobile users are interested in only a very few items. This pattern can be expressed in mathematical format as a power law, meaning that the popularity of the n th most popular category is proportional to $n^{-\tau}$, where $\tau \geq 1$. During each simulation round, we randomly generate sub-populations of mobile users who are interested in each category of data based on Zipf distribution. We adjust these sub-population's interest vectors to reflect this interest distribution. In our simulations, at each simulation round, some mobile nodes are selected arbitrarily as the senders of some data item. The content of this data item is synthesized at the sender in order to represent one of the categories. One restriction on the choice of the sender of a specific data is to select the sender uniformly from the sub-population which is interested in the content of that data. We include this consideration because people usually store data items that they like in their mobile devices. In our simulation scenario, the size of data items is fixed to 500 KB, and we repeat our simulations for low (50 MB), medium (500 MB), and unlimited cache sizes to also study the effect of storage size in the final performance. We chose a transmission budget of 5 MB or 10 data items, per wireless contact.

The allocation of the transmit budget is based on four different methods described in Section 5.2. When two mobile users meet and one of them has some data objects in its cache, those objects are scheduled for transmission at the first portion of the transmit budget if, and only if, their content is of the interest of the other user. At the second portion of the transmit budget, items of other categories are transmitted, if any budget still remains. Our simulations evaluate the performance of the PEEP forwarding algorithm in terms of the speed of distribution of items of interest among mobile users. In other words, we measure the speed of the PEEP algorithm using each of the four different methods of transmit

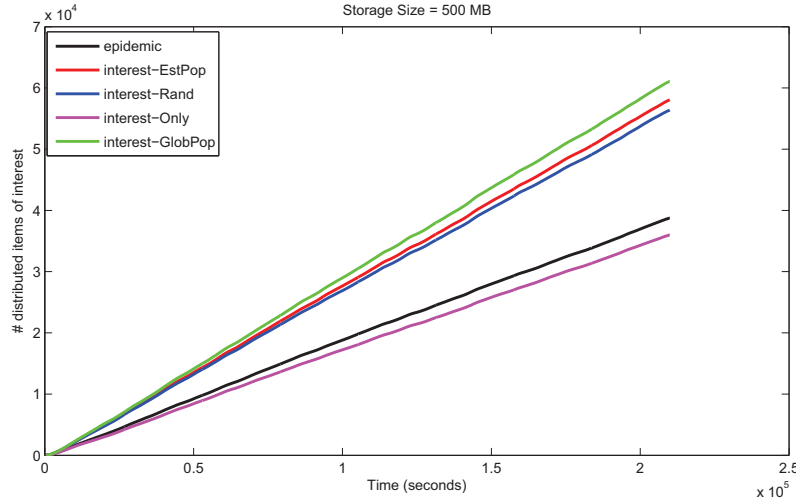


Figure 5.5: Total number of distributed items of interest over time with medium to high cache size.

budget allocation, as discussed in Section 5.2. We also compare the performance of the PEEP algorithm to the epidemic distribution of data items. We have chosen epidemic as our base of comparison for two reasons. First, because, assuming unlimited communication resources such as storage, energy, and bandwidth, epidemic routing is proven to produce maximum data distribution with minimum delay. Second, because all other important routing algorithms which are developed for mobile DTNs do not address data transmission to a group of recipients.

In our simulations, the movement of the mobile users is based on the Random Way Point (RWP) mobility model. The choice of the mobility model characterizes the number and duration of the wireless contacts. Therefore, to have a robust performance that is independent of the mobility model, the normalized rate of the distribution of items of interest should be constant under different choices of the mobility model. In order to verify the robustness of PEEP we also simulate and measure the rate of this distribution scaled in terms of the number of wireless encounters.

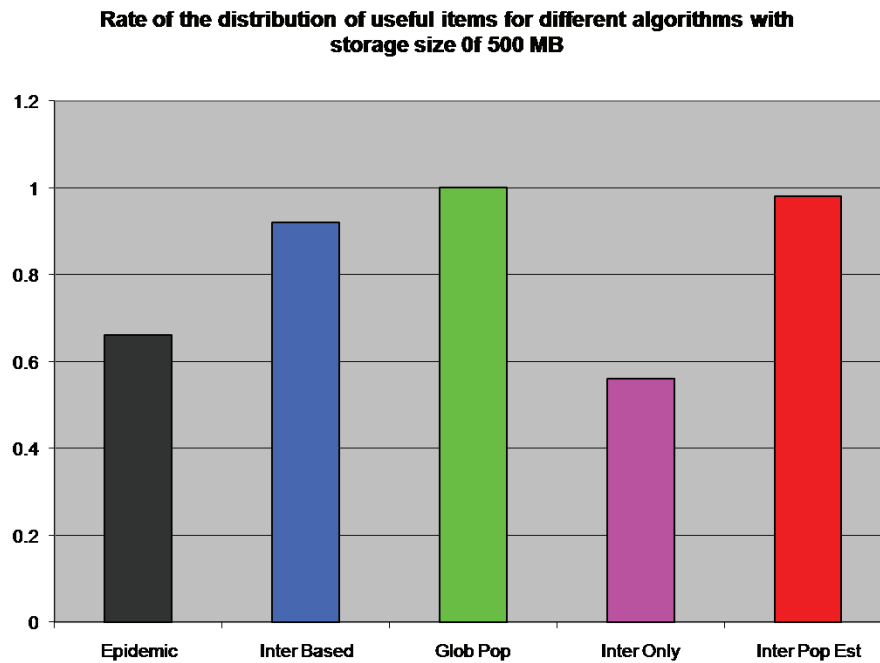


Figure 5.6: Normalized view of the rate of data distribution using different versions of PEEP with moderate cache size (500MB).

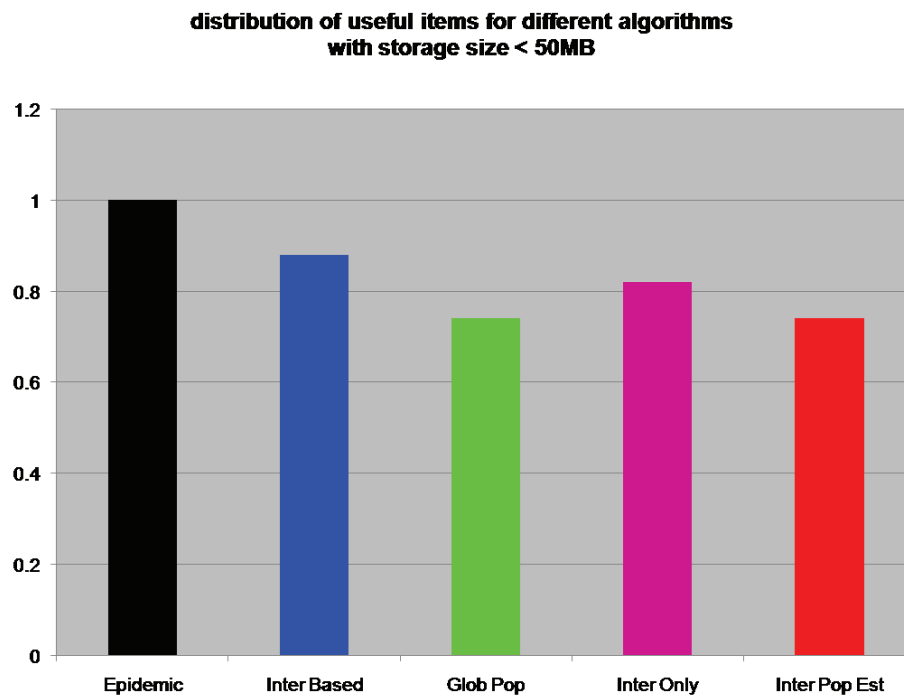


Figure 5.7: Normalized view of rate of data distribution using different versions of PEEP with small cache size (50MB).

5.3.2 Evaluation of the Results

The performance metric in our simulations is the rate of the distribution of items of interest using four different budget allocation methods namely, Interest-Only, Interest-Rand, Interest-Pop, and Interest-GlobPop, as explained in Section 5.2. Figure 5.5 shows the number of items of interest received by the mobile users over time, where mobile devices have a moderate cache size (500 MB in our simulations). The abscissa represents the elapsed time from the start of the simulation in seconds. To decrease the impact of the random data generation in the final simulation results, each data point on the graph has been calculated as the linear interpolation of the 20 trials of the same simulation. We use linear interpolation as a popular method of curve fitting which uses linear polynomials [Meijering, 2002]. As we observe in the graph, the number of received items of interest increases linearly over time. This is the natural outcome of providing more data exchange opportunities for mobile users. Since the graph represents the distribution of items of interest over time, the rate of this distribution can be calculated as the slope of the curves, which is the distinctive feature of each transmit budget allocation method. Figure 5.6 represents the normalized view of the slope of the curves of Figure 5.5. This figure compares the relative performance of different transmit budget allocation methods of the PEEP routing algorithm as well as those of the epidemic routing. As we observe in this figure, the greedy Interest-Only allocation mechanism has the worst performance, even poorer than the epidemic. This outcome contrasts with the simulation results of the interest-aware communication as represented in Section 4.5. This happens because, with the new constraint of transmit budget introduced in PEEP, the data transfer at each wireless contact is limited, therefore slowing down the speed of the overall data distribution. Unsurprisingly, Interest-GlobPop, which is assumed to have a global knowledge about the popularity of data items, provides the highest distribution rate for items of interest. Achieving this global knowledge in an infrastructure-less mobile DTN scenario where mobile users are frequently joining and leaving the network, however, is not practical. But simulation results show that the performance of our local popularity estimation method, as proposed by PEEP's Interest-EstPop, closely follows the optimal performance of the Interest-GlobPop model. Finally, having the third- best performance with the Interest-Rand allocation method verifies that sharing the transmit budget between

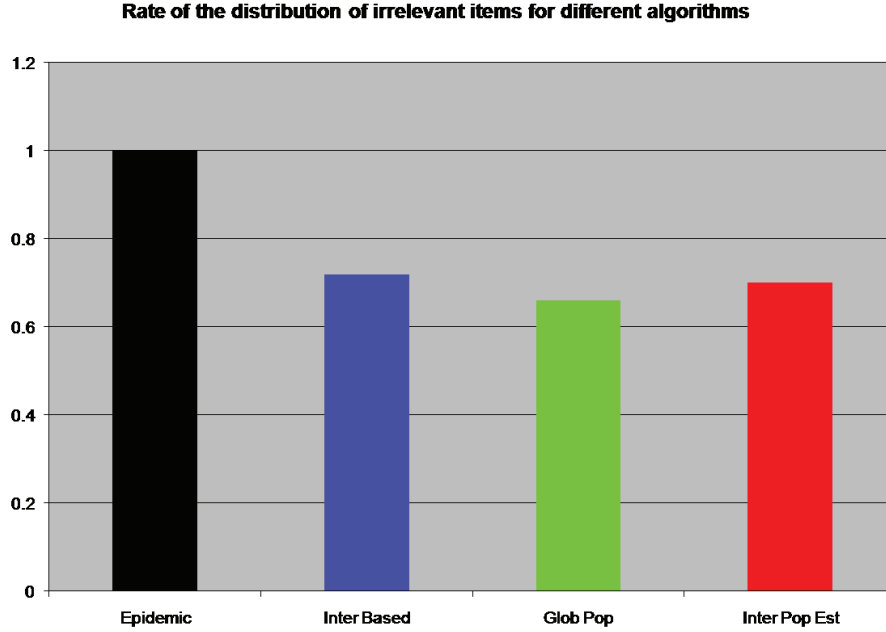


Figure 5.8: Normalized rates of distribution of irrelevant items scaled in terms of encounters.

items of interest and items of other categories yields better performance than the greedy mechanism of transmitting only favorite items.

Since the behavior of PEEP in terms of distribution of items of interest over time is linear, we represent the rest of the figures of this section as normalized bar graphs. Figure 5.7 shows the performance of the PEEP algorithms when the cache size of mobile devices is small (less than 50 MB). As we observe in this graph, all four transmit budget allocation methods have inferior performance to the epidemic multicast, when the cache size is very small. This poor performance happens when the cache size is small and buffer overflow happens too soon and thus items of interest get tossed out and have no chance of distribution beyond few hops. On the other hand, when the cache size is moderate to large (greater than 500 MB), we see a significant improvement in the speed of distribution of items of interest over the epidemic algorithm.

The robustness of the PEEP's algorithm design is verified, if its performance is independent of the choice of the underlying mobility model. Mobility models only characterize the

number and duration of the contacts during the simulation. Therefore, if the normalized rate of data distribution (slope of the graph) measured in terms of the number of encounters remains constant, the performance is independent of the choice of the mobility model. To confirm this fact we repeated our simulations and measured the slopes of the data distribution graphs in terms of the number of contacts. Then, we calculated the normalized values of these slopes. Our final results yielded the same normalized values as Figure 5.6, which verifies the robustness of the PEEP algorithms independent of the underlying mobility model.

Figure 5.8 shows the rate of distribution of irrelevant data items to mobile users' interests by each of the PEEP algorithm's methods as well as by the epidemic multicast. As we observe in this figure, epidemic distributes almost 30% more useless data throughout the mobile DTN compared to any of the PEEP's data distribution methods. Since epidemic multicast has the highest rate of distribution of useless data items and the lowest rate of distribution of items of interest (after greedy Interest-Only), it has the most inefficient way of utilizing communication resources. Epidemic uses communication resources, most importantly energy, to distribute more useless items and fewer items of interest. On the other hand, by referring to Figures 5.6 and 5.8 we realize that Interest-GlobPop and Interest-EstPop algorithms of PEEP distribute more items of interest per wireless contact and fewer irrelevant items. Therefore, they have the most efficient usage of transmission energy at each wireless contact.

5.4 Conclusion

PEEP is an energy efficient content distribution protocol for mobile DTNs that considers user's personal interests as well as the global popularity of data items in its routing decision. PEEP introduces the new concept of transmit budget and proposes different mechanisms to allocate this budget to transmit both locally interesting items as well as globally popular ones. PEEP also proposes a novel but easy way to implement methods to estimate the global popularity of data items by learning from the history of past contacts. When two mobile nodes meet, they first exchange their interest vectors, which represent their individ-

ual interests. Then, based on these interest vectors, they exchange items of interest, and, if there is still transmit budget remaining, they fill the rest of the transmit budget with items of other categories that might be useful for future contacts. We implemented PEEP protocol in the ONE simulator and evaluated its performance in terms of speed of distribution of items of interests. Our simulation results show that our protocol, compared to the epidemic routing and interest-aware models, produces a faster mechanism for diffusion of useful items throughout the network. The PEEP algorithm with popularity estimation method for allocating transmit budget (Interest-EstPop) outperforms epidemic by average of 44%. Furthermore, our simulations verify that PEEP's performance is robust and does not depend on the choice of the underlying mobility model. Moreover, Interest-GlobPop and Interest-EstPop algorithms of PEEP deliver more efficient usage of transmission opportunities and energy, as they transmit more items of interest and fewer irrelevant items per wireless contact.

Part III

Mobility Behavior of Mobile Users

Chapter 6

A Markov Routing Algorithm for Mobile DTNs based on Spatio-Temporal Modeling of Human Movement Data

Store-carry-forward communication, which serves as the heart of all routing protocols for mobile DTNs, exploits nodes' mobility to bring messages closer to their destinations by exchanging messages across mobile nodes whenever they meet in close proximity. Understanding the subtle characteristics of human mobility leads to better service and application provisioning for mobile DTNs. We use GPS traces collected from multiple mobile users to empirically study different aspects of human mobility. Various Markov models (first-, second-, and third-order) are estimated from users' mobility data. Based on empirical evidence, second-order Markov models are deemed sufficient to estimate mobile users' future locations accurately. These Markov models permit the design of a new routing algorithm for mobile DTNs that are capable of more efficiently routing data objects to their destinations. The relay selection in this routing algorithm is based on mobile users' absorption times to the destination location. Simulations show that the proposed routing algorithm consumes less energy than legacy epidemic routing algorithms without excessive transmission delays.

6.1 Introduction

An accurate understanding of human mobility is important for urban planning, traffic forecasting, and monitoring of biological and mobile virus spreading patterns [Kleinberg, 2007]. Human mobility is fundamental to wireless networking research, as it affects other network characteristics such as traffic and connectivity. Mobility impacts the types of applications and routing algorithms that could be provided for mobile networks in general. Knowledge of the mobility and its impact on the performance of the applications and routing algorithms depend on the types of the mobile applications and the environments in which those applications must operate. In cellular networking, for instance, the knowledge of speed and trajectories of mobile users facilitates the design of the cellular hand-off algorithms for QoS purposes. Modeling human mobility allows wireless cellular carriers to provide better service and application provisioning in 3G/4G networks [Schindelhauer, 2006; Sricharan and Vaidehi, 2008]. In mobile DTNs store-carry-forward communication, which is integral to many routing protocols, exploits nodes' mobility to bring messages closer to their destinations by exchanging messages with other mobile nodes when they are in adequate proximity. Therefore, long-term knowledge of future locations of mobile users is a defining factor in the routing of messages to their final destinations in mobile DTNs. In Mobile Adhoc Networks (MANET), on the other hand, short-term knowledge of mobility is required to predict immediate connectivity for end-to-end data transmissions.

Synthetic and trace-based mobility models are two major classes of mobility models that are used to validate new routing algorithms and applications for cellular, ad-hoc, and mobile DTNs. Unfortunately, many current synthetic mobility models do not accurately reflect real-world mobile user behavior [Boudec and Vojnovic, 2006]. In addition, trace-based mobility models are grounded in traces that either represent a specific class of mobile users or are extracted from cell tower traces [Gonzalez *et al.*, 2008] and do not provide sufficient spatial accuracy to capture fine mobility details from users' movement. Such mobility models are primarily used for cellular handoff in cellular networks, i.e. providing a robust protocol that allows movement between cells without interrupting or disturbing communications. Therefore, the human mobility studies were primarily focused on extracting realistic hand-off prediction algorithms for resource management and QoS purposes [Schindelhauer, 2006;

Sricharan and Vaidehi, 2008]. Since these cellular networking algorithms are only interested in movement of a node leaving or entering a cell, they do not require precise location data within the range of any particular cell. Conversely, for infrastructure-less, ad-hoc networks and mobile DTNs, where no cell tower or access point is implicated, fine-grained mobility modeling plays a more crucial role, as data exchange depends highly on device-to-device wireless proximity among mobile nodes. Therefore, the proposed algorithms for mobility predictions for cellular handoffs are not adequate for evaluating mobile ad-hoc and mobile DTN algorithms which require more fine-grained human mobility data. In MANTes or mobile DTNs the choice of mobility traces, synthetic models, and analytic techniques are essential components of any evaluation platform for different algorithms or applications [Abdulla and Simon, 2007; Chaintreau *et al.*, 2007].

Given the aforementioned importance of precision and fine-granularity of real-world traces for mobility models, GPS traces [SenseNetworks, 2008] were collected by tracking a diverse group of mobile users in an urban area. Using these traces, we employ a Markov modeling approach both to investigate different statistical aspects of human mobility and to extract mobile users' mobility patterns. Using this analysis and modeling, we develop a new routing algorithm for mobile DTNs. The routing decision in our Markov based routing algorithm is based on the absorption times of mobile users to different destination locations. Our routing algorithm selects the mobile users with less absorption times to the destination as the data carriers. These absorption times are calculated based on the properties of the Markov chains that we extract from our GPS traces. In addition, due to the probabilistic nature of the Markov models, we use our extracted Markov chains to sample movement data generatively to provide a meaningful simulation environment to evaluate our routing algorithm. Our evaluations show that our proposed Markov based routing algorithm has a more efficient use of energy, as it produces less delay per consumed unit of energy compared to the legacy algorithms. The main objectives of this research are to explore how fine-grained mobility models can improve the performance of routing algorithms in mobile DTNs and thus facilitate service provisioning by wireless carriers in general. We leverage mobility data and the ability to predict mobile users' future locations to develop intelligent data distribution methods in mobile DTNs. For example, software

updates, advertisements, finance news, and other commercial data could be distributed among mobile users more efficiently with such modeling and prediction tools.

This chapter is organized as follows: Section 6.2 briefly reviews previous work and discusses the inadequacies of mobility models currently being used to design and evaluate routing protocols. Section 6.3 describes the characteristics of our database of movement traces [SenseNetworks, 2008] that distinguish it from previously introduced mobility traces. Section 6.4 explains the details of the mobility models that can be extracted from the location traces. Section 6.5 focuses on second-order Markov models in particular, which seem especially well-suited to the movement data sets being considered. Section 6.6 details the design, implementation, and evaluation of a Markov based routing algorithm for mobile DTNs. Finally, Section 6.7 concludes this chapter.

6.2 Related work

There are two classes of mobility models that are used in simulations of mobile networks. These models are either produced synthetically, or they are based on real-world traces. Synthetic mobility models are usually based on randomized recreation of the movement of mobile users. Trace-based models are used either to set the parameters of the synthetically created random models or as actual traces to represent the population's behavior. In this section we describe the insufficiencies of each of these models.

6.2.1 Synthetic mobility models

Many routing algorithms and applications that are designed for mobile ad-hoc networks or DTNs are validated through simulations. [Kurkowski *et al.*, 2005] has concluded that almost 76% of Mobihoc papers used simulations to evaluate their results, and all simulation-based studies use some class of mobility models in their experiments. Research verifies that the performance results of an ad-hoc network protocol change drastically as a result of changing the mobility model in the simulation [Camp *et al.*, 2002].

All synthetic mobility models are either a variation of the random way point (RWP) mobility model, or they have been implemented by modifying some parameters of the RWP

model [Camp *et al.*, 2002]. In this model each node moves from its current location to a new location by randomly choosing an arbitrary direction and speed from a given range. Such a move is performed for either a constant time or for a constant distance. Then, a new speed and direction are chosen. Some of these synthetic models try to add some realistic assumptions to the RWP model. For example, SWIM [Mei and Stefa, 2008] is based on RWP but adds the restriction that mobile users travel most often to places close to their home. The community-based mobility model [Musolesi and Mascolo, 2006; Hu and Dittmann, 2009] is based on grouping mobile users and assigning them to different locations based on their social relationships with others. This model is oblivious to the fact that social interactions are not the only defining factor of human movements. Working day movement model [Ekman *et al.*, 2008] and city section mobility model [Davies, 2000] emulate the RWP model based on the street map of a city. In the Gauss-Markov mobility model [Liang and Haas, 1999], a node's next location is generated by its past locations and velocity. In the probabilistic version of the RWP model [Chiang, 1998], the last step made by the random walk influences the next one based on some probability distribution. The main problem with these synthetic mobility models is their simplified assumptions which make the model more tractable and easier to implement. This leads to significant deviation of the simulation results from the real-world implementation of the protocols and routing algorithms for ad-hoc networks and mobile DTNs. For instance, the most common problem with simulation studies using the random way point model is a poor choice of velocity distribution [Yoon *et al.*, 2003], e.g., uniform distribution $U(0, V_{max})$. Such velocity distributions (which seem to be common with NS-2 simulations) lead to a situation where each node stops moving at the stationary state.

6.2.2 From real-world traces to mobility models

Because synthetic mobility models are inadequate to reflect users' movements in the real-world scenarios, exploring characteristics of human mobility by using real-world mobile traces becomes important. Therefore, trace-driven simulations are a more realistic alternative to evaluate the performance of different mobile DTN algorithms and applications. Although trace-driven simulations do not require a mobility model, a model-based simu-

lation allows researchers to explore different scenarios in their simulations by providing a larger parameter space. Different studies have been conducted to extract mobility models from real-world movement traces. A summary of these studies and different movement traces they use to extract different trace-driven mobility models are represented in Table 6.1. We explain the details of the projects that are highlighted in this table as follows.

Minkyong et al. [Mikyong *et al.*, 2006] have derived the distribution of speed and pause times of mobile users using the WiFi access point traces collected at Dartmouth College [Henderson *et al.*, 2004]. Based on their trace-driven calculations, both speed and pause time follow a log-normal distribution. Rhee et al. [Rhee *et al.*, 2008] have studied human walks based on GPS traces involving 44 volunteers. They show that many statistical features of human walks, such as distribution of consecutive displacements and pause time distributions, follow truncated power-laws. They show that these traits are similar to those of Levy walks [Shlesinger *et al.*, 1982]. In another study Gonzalez et al. [Gonzalez *et al.*, 2008] used the mobility traces of 100,000 individuals over a six-month period that were logged into cell towers in order to analyze human mobility distributions. By investigating these traces, they also suggest that human motion follows a truncated Levy flight, which is a random walk for which step size follows a power-law distribution.

Although the statistical results of these mobility models are significant, they are not adequate. One problem with these models is their precision, as they have been extracted either from WiFi [Mikyong *et al.*, 2006] or cell tower traces the granularity of which is as large as the WiFi AP's or cell tower's range [Gonzalez *et al.*, 2008; Song *et al.*, 2010]. Another problem with some of these models is their inference from traces that characterize either a specific class [Mikyong *et al.*, 2006] or a small number of mobile users [Rhee *et al.*, 2008]. Therefore, generalizing these models to a more general subset of the human population has yet to be achieved. Brockmann et al. [Brockmann and Theis, 2008] have generalized Levy statistics to humans, documenting that the distribution of distances between consecutive sightings of nearly half-a-million bank notes is fat-tailed power-law. However, because each consecutive sighting of bank notes reflects the composite motion of two or more individuals, it is not clear whether the observed distribution reflects the motion of individual users. Girardin et al. [Girardin *et al.*, 2008] have used a method to uncover the presence and movements of tourists

from the geo-referenced photos which the users generate and post on flickr. The existence of patterns in human movement could facilitate service provisioning in mobile networks for a wide range of applications from prevention of human and electronic viruses to resource management in mobile communications. Therefore, the recent research efforts in human mobility are focused on the possibility of forecasting the mobility of individuals [Wang *et al.*, 2009; Eagle and Pentland, 2006b]. By measuring the entropy of each individual's trajectory, Song *et al.* [Song *et al.*, 2010] found a 93% potential predictability in user mobility across the whole user base. Gonzalez *et al.* [Gonzalez *et al.*, 2008] also conclude that human movement follows certain patterns, as users tend to return to locations that they have visited before. Our goal in this chapter is to characterize the inherent patterns in mobile users' movement and to use this knowledge to design a more intelligent routing algorithm for mobile DTNs.

SOLAR [Ghosh *et al.*, 2007] routing algorithm for mobile DTNs bases its routing decision on delivery probability of each mobile node to the destination. The delivery probability for each mobile user is calculated by extracting "mobility profile" of that user which is the frequency of previous visits to a set of predetermined hot-spots or hubs. SOLAR tries to route data to these hubs by selecting relays which have higher probability of visiting those hubs. In our algorithm, however, pattern has a broader definition and is not limited to visiting a set of predetermined hubs with a predefined, constant size. Furthermore, SOLAR's performance has been evaluated using RWP mobility model, while we design and evaluate our algorithm using movement traces which are extracted from real-world mobile users.

6.3 Characteristics of our mobility traces

Because of the importance of the mobility models in regenerating different mobility scenarios for simulations, we develop a new mobility model, which is extracted from real-world mobile users' movement traces within a broad spatial and temporal spectrum. We first explain the unique characteristics of these traces and then describe the statistical techniques we use to extract our new mobility model.

We are using the movement traces collected by Sense Networks [SenseNetworks, 2008]

in our mobility model studies. Sense Networks has created a research platform that utilizes GPS to gather traces of a diverse class of mobile users. In order to encourage users to contribute their movement data, Sense Networks has implemented a mobile social networking application called Citysense [Citysense, 2008]. Mobile users can download the Citysense application on their mobile devices to discover local nightlife and to participate in social navigation by answering the question, "Where is everybody?" By using this application, users are able to monitor the overall activity level of the city, top activity hotspots, and places with unexpectedly high activity all in real-time. Moreover, users can locate their friends using the buddy-finder application. By installing the Citysense application, mobile users agree to contribute their own geo-location data to the Sense Networks' database. To address privacy concerns, Sense Networks has developed a new approach to data ownership by mobile users. Users are the primary owners of any historical data that they create within the system, and they can choose to delete their data anytime they wish. By offering personalized services and the guarantee of privacy protection, Citysense encourages more people to use the application. This approach increases the chance of attracting a wider range of mobile users to use the application and contribute their geo-location trace to the system. In fact, Sense Networks is creating a database of as many as 10,000 users from major cities in the U.S. These GPS traces are collected from pings that are sent to mobile devices every half-hour with the precision of 20 feet (GPS pings precision) which guarantees the temporal and spacial granularity of these traces compared to the traces collected from APs or cell towers. Device-to-device contact traces, just like the reality-mining database [Mining, 2003], can also be generated from the GPS traces by calculating the distance of the mobile users from each other, the time of the contact, and a consideration of the range of wireless interfaces.

6.3.1 Extracting different features of human movement

The first step in analyzing the movement traces is representing mobile users' GPS traces in a meaningful format. Our representation of the movement data is based on focusing on certain geographical areas, here northern California. As shown in Figure 6.1, we have divided this area into smaller activity areas or grids of size 300 ft \times 300ft which is as

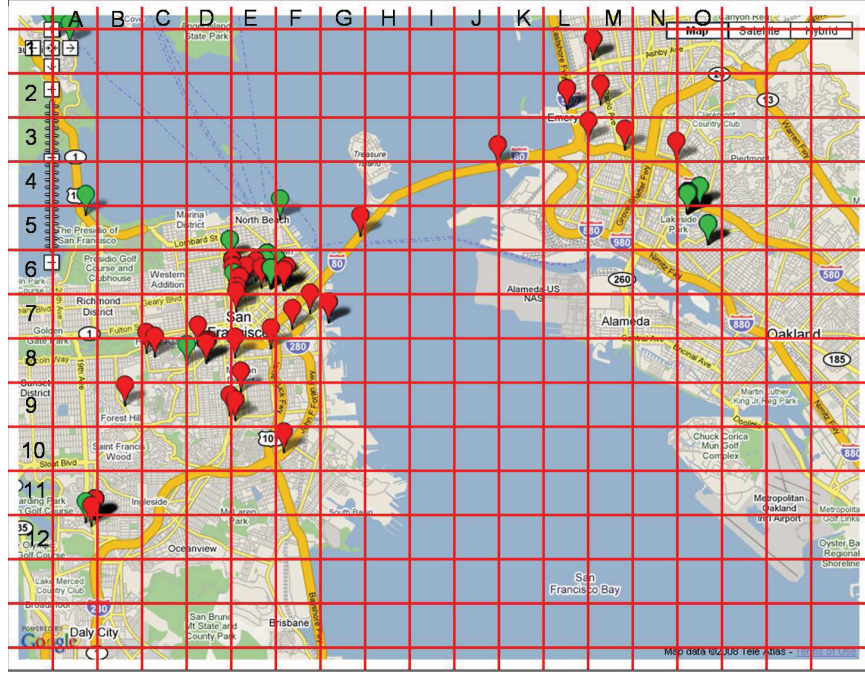


Figure 6.1: Map grid of San Fransisco with number of GPS pings at each grid.

twice as the range of the WiFi wireless interface, to guarantee the possibility of wireless communication between mobile users in the same grid. Thereby, we are excluding cases where mobile-to-mobile communication is not possible from our consideration. Each mobile user has a certain mobility profile. For example, some mobile users are random roamers and tend to visit many different grids in a short amount of time, while other mobile users have a more limited activity area, spending most of their time in very few locations. In contrast with the random trajectories predicted by the Levy flight and random walk [Boudec and Vojnovic, 2006], human trajectories show a high degree of temporal and spatial regularity, meaning, humans tend to revisit the locations they have visited in the past [Gonzalez *et al.*, 2008]. Generally, most users tend to visit a small number of locations such as home, work, and neighborhood grocery shops more frequently than other locations. There are also some other users who have more unpredictable patterns of movement, for example, taxi drivers. These users are classified as random roamers. Many researchers have tried to analyze the existence or nonexistence of such patterns in human movement. We define the existence of regularity or pattern as the existence of few locations that are visited more frequently

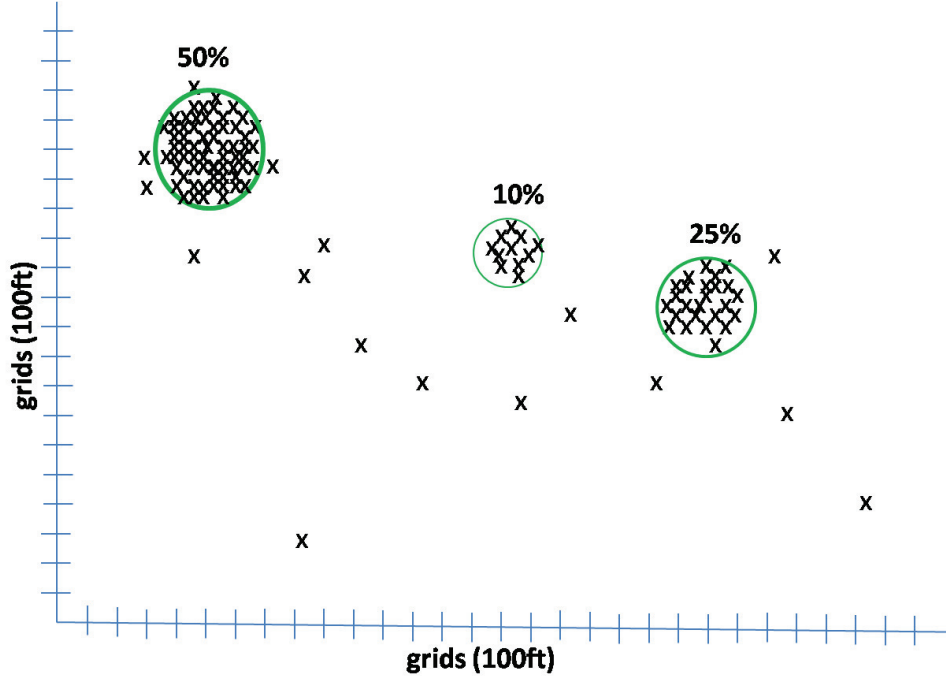


Figure 6.2: A conceptual illustration of fitting Gaussian distributions on a user's movement data. In this example the user has 3 predominant locations where she spends 50%, 10%, and 25% of her time.

by mobile users. These patterns in human movement can be identified by calculating the prominent locations where mobile users spend most of their time (e.g. 85% of their time).

As we explained earlier, we divide users' movement maps into different regions or grids of activity. Users spend some percentage of their time in each region based on their own mobility profiles. This percentage can be also interpreted as the probability of the user being present at that specific location. We first identify the predominant activity locations of each user by fitting Gaussian distributions on individual users daily movement data as shown in Figure 6.2. Based on our traces, for the majority of the mobile users, 15% of the GPS pings on average are randomly scattered across the users movement map, and, therefore, they do not contribute to the pattern extraction. This Gaussian fitting verifies that the majority of human subjects in our sample population spend, on average, more than 85% of their time in a few locations. The number of these predominant locations are few, less than 6, during workdays and a maximum of 15 locations during weekends for the majority

of the users. We also identified some random roamers with more uniformly scattered GPS pings across a large number of regions. The number of these users is less than 10% of our sample population, and we exclude them from our pattern extraction analysis.

The percentage of the time each user spends at each location and the number of prominent locations depends on the latent factors in human mobility. For example, the type of the day (such as a workday or a weekend) or the type of profession differentiates the mobility of users who spend most of their weekdays in an office from the mobility of a taxi driver who is categorized as a random roamer. The fact that the majority of our sample population spends more than 85% of their daily time in a few locations is correlated with the demography of the people who carry blackberries, as they tended to be mostly office workers at the time of this research.

6.4 Modeling mobile users' movements by N-grams

The design goal for our movement model is to determine the possibility of predicting each user's most probable future locations after learning their past behavior. Our model calculates users' daily movement behavior for individual days of a week. Learning behavior for individual days of a week is equivalent to defining a time window of one day to learn users' movement. The size of this time window in temporal analysis of mobile users' movement depend highly on the type of application. For example, in traffic management applications a meaningful time window to learn users' behavior should be narrowed down to rush-hours and regular hours of the day. Even for more personalized mobile applications a user might have different movement patterns across different hours of a day, for example, a user might spend the entire 9am to 5pm time window in her office, while having a totally mobile lifestyle after work hours. The level of granularity in the time window that we have considered in our movement model is days of the week. This means that we learn and model each user's movement for each day of the week based on the traces collected for that day across multiple weeks. We make this assumption about granularity in time without any loss of generality in our model. The statistical methods that we implement as part of our model could be supplied with different traces of data and with different levels of granularity to produce the

corresponding output correctly.

In our model human movement is represented as a sequence of consecutively visited grids. Each element of this sequence is the grid in which the user was located at the time of the GPS ping. Using this representation, the problem of predicting a user's future move converts to the problem of predicting the next item in the sequence. N-grams that are used in various areas of natural language processing (NLP) and genetic sequence analysis are a type of probabilistic approach for predicting the next item in sequences of words or DNA. We have adopted n -grams in modeling the sequence of human moves.

An n -gram is a sequence of n items. The items in sequence can be phonemes, syllables, letters, words, or grids, according to the application. An n -gram of size 1 is referred to as a *unigram*; size 2 is a *bigram*; size 3 is a *trigram*; and size 4 or more is simply called an *n-gram*. The models built from n -grams are (n) -order Markov models.

Based on this definition, in unigram modeling of human movement, future locations of the users are estimated independently from the knowledge of users current or past locations. Therefore, the probability of finding a user in a specific location is simply calculated as the percentage of the time that that specific user spent in that location in the past. The percentage of the time that users spend in each location is learned from the users' movement traces. The unigram model is a naive way of modeling the movement behavior, as it does not capture any trajectory in the human mobility, while intuitively we expect users to cross the grids one after the other when they travel. Therefore, a user's next location should depend at least on the user's current location. This observation leads us to use the bigram model, which is in fact a subset of second-order Markov models or Markov chains.

Although using history seems intuitively to be helpful in learning human movement, it is not immediately clear how far in the past we should examine the locations in order to produce more accurate future predictions. Therefore, we also study trigram models of users' movement, which, as well as the current location of the user, takes into account one step back into the past. This is actually the same as using the third-order Markov models to calculate the likelihoods of future locations by considering two consecutive previous locations: the current location and the one before the current location.

Tuples of Grids	5039907665	5038663466	5038414624	5038414623	5060063904	5053345115	...
5039907665	1370	10	230	10	0	30	...
5038663466	30	130	110	0	0	0	...
5038414624	220	110	3420	120	0	60	...
5038414623	10	0	50	0	14	0	...
5060063904	0	0	12	110	0	0	...
5053345115	0	50	0	13	176	343	...
.
.

Figure 6.3: A snippet of a bigram table that is calculated from a user’s movement data during a weekend. Bigram tables are calculated from the 80% portion of the movement data which is considered as train data.

6.4.1 Bigram statistics

Human movement traces can be converted to a large sequence of consecutively visited grids. In our model each element in this sequence corresponds to a grid number where a user was located at the time of the GPS ping. Therefore, transition from one element in the sequence to the next occurs every 30 minutes, which is the time elapsed between two pings.

We consider a larger subset of each user’s movement trace, here 80%, as our training data. We implement a program to learn the user’s bigram table, in other words, to train the bigram model from the training data. This program creates a large table of all grid bigrams found in the body of the user’s daily movement trace. The program scans the train sequence element by element, like a sliding window, across the trace from the beginning and stores two grid sequences as it moves along until the last grid in the train dataset is reached. On the output, it creates the bigram table, which stores the frequency of observing tuples of two consecutive grids in the trace. Figure 6.3 shows a section of a bigram table as an example.

In order to learn the daily movement patterns of a user, we bundle all traces of the user's daily movement data across multiple weeks. The logic behind this categorization of data is that we assume the majority of the users manifest similar movement behavior on similar weekdays, for example, on all Mondays. Then, for each data trace corresponding to each day of the week, we use 80% of the data to train the bigram model and the remaining 20% to test the bigram model. Training the bigram model is the same as creating the bigram tables, and testing the model is equivalent to calculating the likelihood of the observed sequence in the test data.

6.4.1.1 Calculating Likelihood of the Movement Trace

We calculate the likelihood of a test movement trace by referring to the bigram tables. First, we assume the observed test sequence is in the form of:

$$G_1, G_2, G_3, \dots, G_i, G_{i+1}, \dots, G_N$$

Calculating the likelihood is calculating the probability of this observed sequence as follows:

$$P(G_1, G_2, G_3, \dots, G_i, G_{i+1}, \dots, G_N) = P(G_1) \times P(G_2 | G_1) \times \dots \times P(G_{i+1} | G_i) \times \dots \times P(G_N | G_{N-1})$$

The elements of this multiplication are calculated by referring to the corresponding rows of the bigram table.

$$P(G_{i+1} | G_i) = \frac{P(G_i \wedge G_{i+1})}{P(G_i)}$$

$P(G_i \wedge G_{i+1})$ is the element of the bigram matrix located at row i and column $i + 1$, which is the frequency of observing tuple (G_i, G_{i+1}) in the training data, divided by the total number of items in the train data. $P(G_i)$ is the sum of all columns of row i divided by the total number of items in the train data.

$P(G_i)$ is the unigram probability of finding the user in grid i . Hence, unigram statistics can also be calculated from the bigram matrix. Consequently, the unigram likelihood of observing sequence $(G_1, G_2, G_3, \dots, G_i, G_{i+1}, \dots, G_N)$ is calculated as the following:

$$\mathcal{L}(G_1, G_2, G_3, \dots, G_i, G_{i+1}, \dots, G_N) = P(G_1) \times P(G_2) \times \dots \times P(G_i) \times \dots \times P(G_N) \quad (6.1)$$

Since the probability of an event happening is always less than or equal to one, the multiplication in the Eq. (6.1) of N probability terms produces very small numbers. Therefore, in order to have a meaningful comparison among these small values, we calculate log-likelihoods instead of directly using the probability in our statistics. This turns the multiplication into a summation:

$$\log \mathcal{L}(G_1, G_2, G_3, \dots, G_i, G_{i+1}, \dots, G_N) = \sum_{i=1}^n \log P(G_i)$$

6.4.2 Trigram statistics

In our model we refer to trigram as a triple of grids. Trigram modeling is used in statistical natural language processing (NLP) to calculate the probability of observing a specific word after two other preceding words. We use this method in our model as the probability of a user visiting a specific grid after visiting two preceding grids, and we write this probability as $P(G_i \mid G_{i-1}, G_{i-2})$.

Trigram tables are created in a similar mechanism as bigram tables from the train dataset. However, this time the algorithm for scanning the train dataset counts for trigrams (triples) instead of bigrams (tuples) and stores their frequency inside the trigram table. Here, each tuple of grids indexes a row in the trigram table and the following grid after the tuple indexes the corresponding column in that row. A partial view of one of the trigram tables of a mobile user is represented in figure 6.4.

After extracting the trigram tables, the likelihood of the test movement trace is calculated as follows.

$$\begin{aligned} \mathcal{L}(G_1, G_2, G_3, \dots, G_i, G_{i+1}, \dots, G_N) &= P(G_1, G_2, G_3, \dots, G_i, G_{i+1}, \dots, G_N) \\ &= P(G_1) \times P(G_2 \mid G_1) \times P(G_3 \mid G_2, G_1) \times \dots \\ &\times P(G_i \mid G_{i-1}, G_{i-2}) \times \dots \times P(G_N \mid G_{N-1}, G_{N-2}) \end{aligned}$$

Triple of Grids		5039907665	5038414624	5050607875	5053345115	5038414623	...
5039907665	5039907665	1180	110	30	30	10	...
5039907665	5038414624	0	230	0	0	0	...
5038414624	5038414624	220	2820	10	60	110	...
5038414624	5039907665	110	100	0	0	0	...
5039907665	5050607875	0	20	10	0	0	...
5050607875	5038414624	0	30	133	0	44	...
.
.

Figure 6.4: A snippet of a trigram table that is calculated from a user’s movement data during a weekend. Trigram tables are calculated from the 80% portion of this movement data that is considered as train data.

Each $P(G_i | G_{i-1}, G_{i-2})$ item of this equation is calculated by referring to the trigram table in a similar method as that which we explained for the bigram statistics.

6.4.3 Evaluation of the N-gram models

The question we are trying to answer in this section is how accurately these n -gram models explain patterns in human movement. As we explained before, intuitively we expect bigram models to be more accurate than unigrams as they take into account the inherent trajectories in the mobile users’ movement. However, whether trigrams are more accurate than bigrams and unigrams is not immediately obvious. In other words, by calculations that are derived from real-world data, we determine what order of Markov models explains the human movement better.

Figure 6.5 represents log-likelihoods of test movement traces of a subset of our population, consisting of 31 users, that are calculated after learning their unigram, bigram, and trigram models. We first train the n -gram models based, on each user’s train data. Then

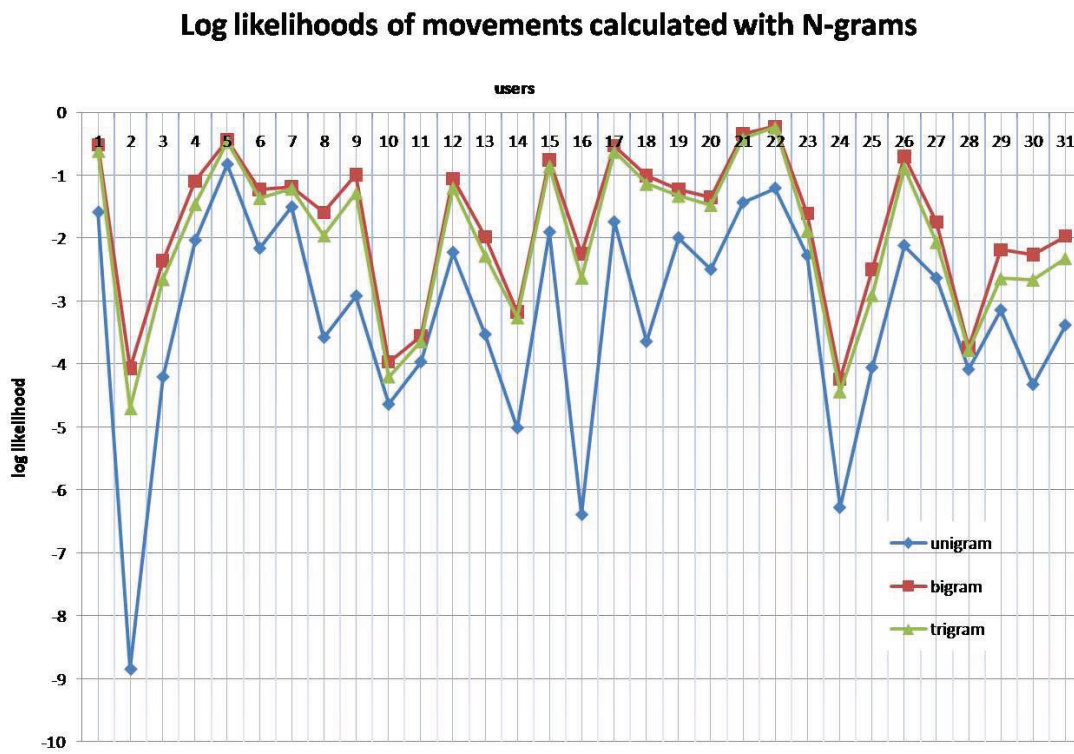


Figure 6.5: Log likelihoods which are calculated for each user’s test data based on their workday n-gram tables.

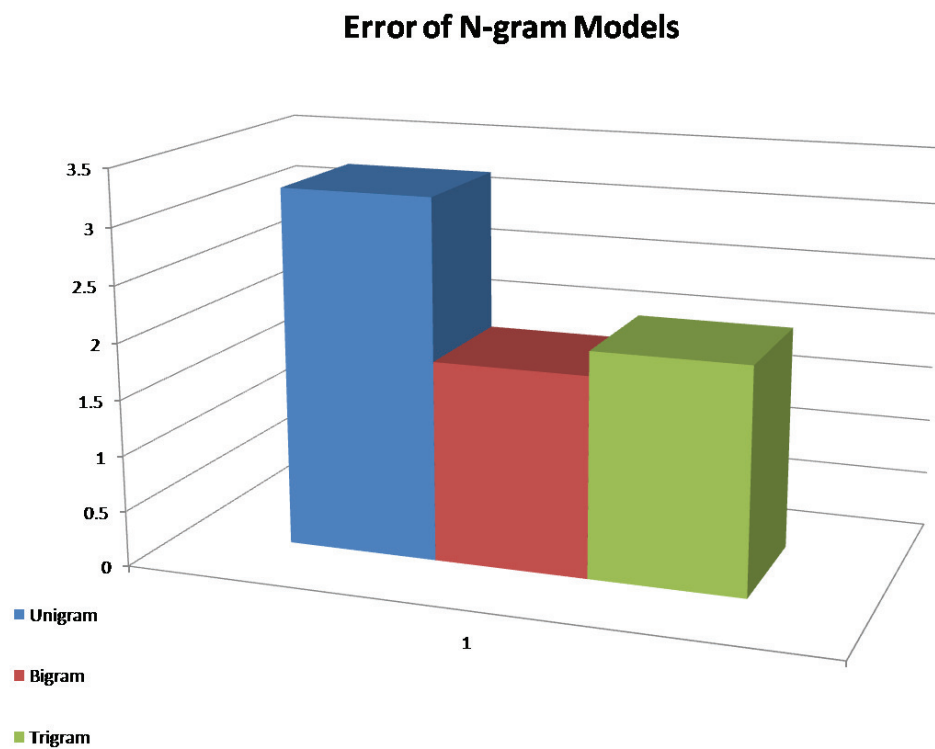


Figure 6.6: Comparison between different n-gram models based on their average errors in calculating log likelihoods of users' test data.

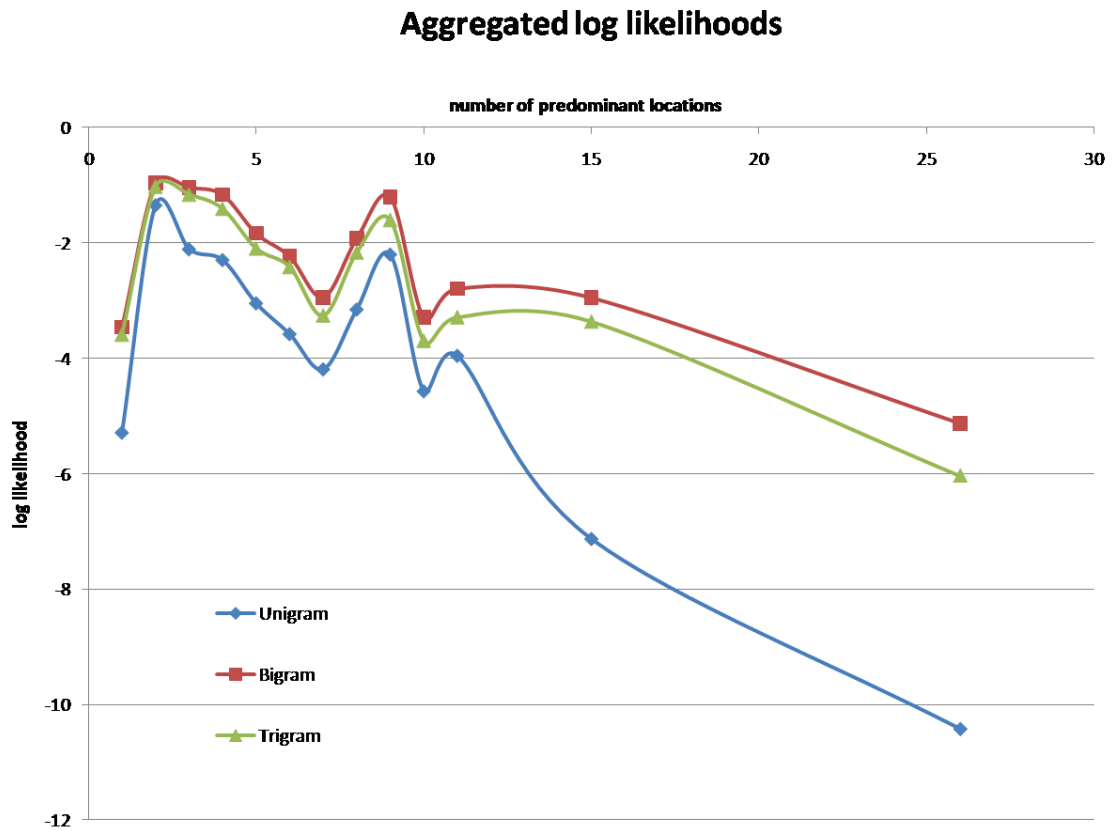


Figure 6.7: Log likelihoods which are calculated for aggregated users' data from Figure 6.5. In this figure users are classified together based on their predominant locations.

we calculate the log likelihood of the test data for each of the n -gram models, as explained in the previous sections. Since the log-likelihoods are negative we define the amount of the error produced by each model to be the negative of its log-likelihood. As we observe in Figure 6.5, the unigram model which does not consider any history to predict users' future locations produces more error which is expected. Moreover, as we can see, bigram and tri-gram models follow each other very closely in terms of the produced error, and also they are both on average 40% more accurate than the unigram model. The average produced error by each n -gram model is shown in Figure 6.6. Figure 6.7 displays the aggregated statistics of Figure 6.5 pivoted on the number or prominent visited locations by mobile users.

Since bigram and trigram produce similar error values and also because trigram adds one more degree of complexity to the model without any significant gain over bigram, we use bigram model in the rest of our analysis to model mobile users' movement. The bigram model is conceptually the same as the second-order Markov model or Markov chains as it considers the current state to predict the future states. Therefore, we model human mobility by Markov chains and we investigate different characteristics of this model in the next section.

6.5 Specifying a Markov chain for human movement

Based on our previous discussion, defining and evaluating the n -gram tables led us to study the patterns in human movement by Markov chains. A Markov chain is a discrete random process with the Markov property. A discrete random process means a system which is in a certain state at each "step", with the state changing randomly between steps. The steps are often thought of as time, i.e., time between consecutive GPS pings in our system. Users might change the state by moving from one grid to another one between two GPS pings by walking, driving or any other means.

We describe a Markov chain for each individual user's movement as follows: we define a set of locations or states, $S = \{s_1, s_2, \dots, s_r\}$. The user starts in one of these states or locations and transitions successively from one state to another with some probability. Each transition corresponds to each GPS ping and is called a step. If the chain is currently in

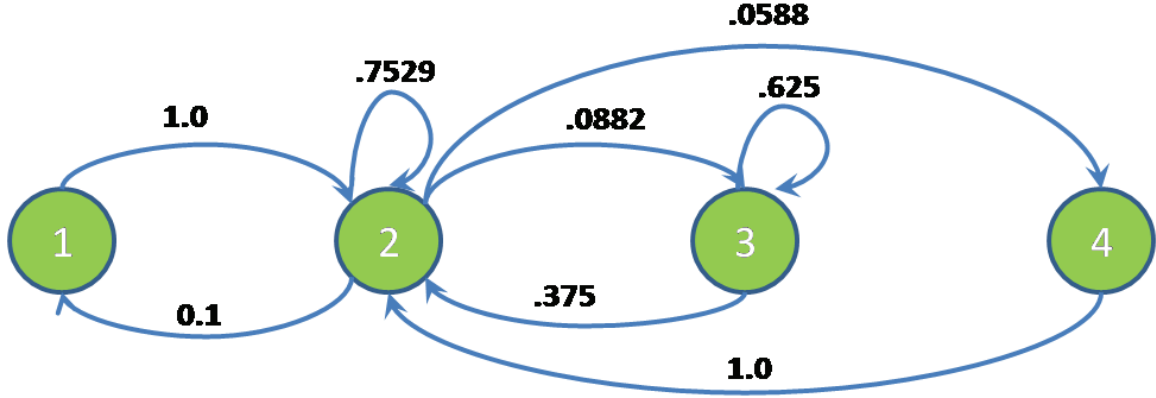


Figure 6.8: An example of a user’s Markov chain which is extracted from a user’s workday movement trace. This Markov chain is also ergodic as higher powers of its transition matrix converges to a stationary distribution.

state s_i , then it transitions to state s_j at the next step with a probability denoted by p_{ij} where j could also be equal to i , which means that the user stays at the same state. The probability p_{ij} indicates the frequency of a user’s transitions from state s_i to the state s_j . Based on the definition of Markov chains, this probability does not depend on the states the chain was at before the current state, i . The Markov chain is defined with the resulting transition matrix P , with p_{ij} as its elements. Transition matrix P describes each user’s daily movement profile. This movement profile can be used to classify users in different movement categories. Figure 6.8 shows a sample Markov Chain calculated from one user’s movement database on Monday. In the following sections we first enumerate the important properties of the Markov chains which are extracted from users’ movement traces. Then, we design a routing algorithm that uses users’ mobility profiles, which are represented in the form of transition matrices to select the most suitable forwarders to route data to a destination location.

6.5.1 Characteristics of the Markov chain

Our goal is to investigate the possibility of existing patterns in human movement. One necessary condition for the existence of any pattern is the recurrence of an incident. The patterns in human movement can be interpreted as indication of whether humans revisit

the locations they have visited before. In order to deduce any such properties from mobile users' movements, we first need to study the characteristics of the Markov chains we derive from mobile users' movement traces.

6.5.1.1 Ergodicity of the Markov Chain

In the mathematical sciences, a stationary process (or strictly stationary process or strongly stationary process) is a stochastic process whose joint probability distribution does not change when shifted in time or space. As a result, statistical properties such as the cumulative distribution function (CDF), the mean, and the variance, if they exist, also do not change over time or position. Claiming that human movement is strictly stationary means with any shift in time the process (CDF, means and variance) should stay consistent. In order to prove the human movement is statistically stationary we should have data that extends across a broad spectrum in time, which is practically impossible at the time of this research. The data that we have based our analysis upon is rather *seasonal* or in another word, a snapshot of human movement in time. Therefore, here we emphasize the necessary conditions for this process to be stationary, and we explore whether our seasonal movement data sustains those conditions.

If a Markov chain is stationary or time-homogeneous, so that the process is described by a single, time-independent matrix p_{ij} , then the vector π is called a stationary distribution (or invariant measure), if its entries π_i sum to 1 and if it satisfies:

$$\pi_j = \sum_{i \in S} \pi_i p_{ij}$$

Furthermore, the chain converges to the stationary distribution regardless of where it begins. Such π is called the equilibrium distribution of the chain. This stationary distribution can be calculated as:

$$\pi_j = \lim_{n \rightarrow \infty} p_{ij}^{(n)}$$

The class of Markov chains that have a unique stationary distribution is called *ergodic* [Izquierdo *et al.*, 2009]. A state is said to be ergodic if it is aperiodic and positive recurrent. If all states in a Markov chain are ergodic, then the chain is said to be ergodic.

1. Periodicity:

A state i has period k if returns to state i occur in multiples of k time steps. Formally, the period of a state is defined as:

$$k = \gcd\{n : \Pr(X_n = i \mid X_0 = i) > 0\}$$

where gcd is the greatest common divisor. It should be mentioned that even though a state has period k , it may not be possible to reach the state in k steps. For example, suppose it is possible to return to the state in 6, 8, 10, 12, ... time steps; then k would be 2, even though 2 does not appear in this list. If $k = 1$, then the state is said to be aperiodic, i.e. returns to state i can occur at irregular times. Otherwise ($k > 1$), the state is said to be periodic with period k .

2. Recurrence:

A state i is said to be transient, if given that we start in state i , there is a non-zero probability that we will never return to i . Formally, let the random variable T_i be the first return time to state i or "hitting time" of state i :

$$T_i = \inf\{n \geq 1 : X_n = i \mid X_0 = i\}$$

Then, state i is transient if and only if:

$$\Pr(T_i = \infty) > 0$$

If a state i is not transient (it has finite hitting time with probability 1), then it is said to be recurrent or persistent. State i is positive recurrent, if it has a finite expected return time, $E[T_i] < \infty$, otherwise, state i is null recurrent. Positive recurrence of our Markov chain which is derived from the human movement trace is a necessary condition for existence of a pattern in human movement, because positive recurrence implies that humans revisit the locations they visited before.

Since ergodicity of the Markov chain is a necessary and sufficient condition for the chain to have stationary distribution and for the existence of a pattern in human movement, it is

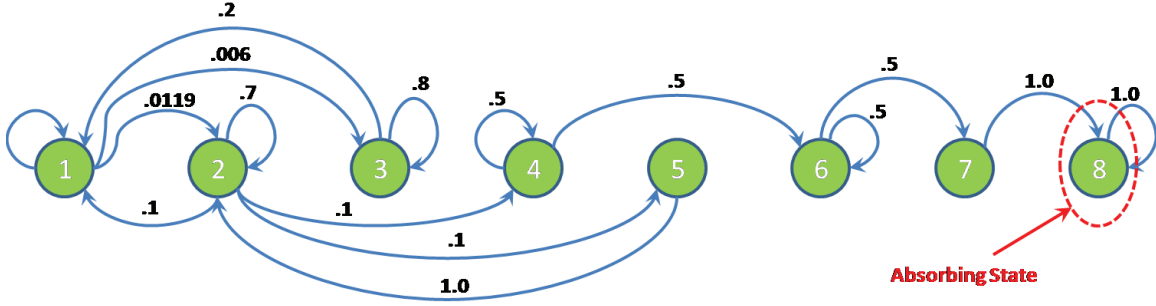


Figure 6.9: An example of a user’s Markov chain which is extracted from a user’s workday movement trace. This Markov chain turns out to be nonergodic as higher powers of its transition matrix does not converge to a stationary distribution.

important that this property be investigated, using the real-world mobility traces. Here, in order to examine whether human movement is ergodic, we investigate if the Markov chains that are extracted from mobile users’ traces are ergodic. In other words, we check to see if rows of the transition matrices converge to a stationary distribution, π .

6.5.1.2 Ergodicity of Human Movement

In order to investigate the ergodicity of human movement, we calculated the daily transition matrix for each individual user and evaluated its convergence over time. In our analysis we detected two types of Markov chains from users’ traces. The first type is extracted for users who have large movement traces in the database that extends over a long period of time. This type of Markov Chain turns out to be ergodic, and an example is represented in Figure 6.8. The second type of Markov chain, however, does not converge to an equilibrium state. After further investigations we realized that these types of chains contain some absorbing state. An absorbing state is a state from which there is a zero probability of transition to another state. After further investigation, we realized the main reason for existence of these absorbing states is the abrupt termination of the GPS pings at a certain location. The traces show that these Markov chains belong either to the users who have a short movement history in the database or to the users who have decided to erase certain parts of their traces from the database. An example of such entries has been shown in Figure 6.9.

6.6 A Markov based routing algorithm for mobile DTNs

In this section we introduce a routing algorithm for mobile DTNs based on our second-order Markov analysis of human mobility to route a data object to a specific location. Most of the routing algorithms for mobile DTNs are designed to route data from a source to a specific destination where this destination is usually another mobile user [Arezu Moghadam and Henning Schulzrinne, 2009]. There are scenarios, however, in which data needs to be routed to a group of mobile users located in a specific location rather than to a certain individual. In these scenarios, mobile users need to be informed of an event based on the region in which they are located. This can occur, for example, when an emergency notification needs to be transmitted to mobile users in a certain geographical location or in a social networking scenario, when friends who are hanging out in a certain part of the city need to be notified by other friends about a particular social event in that neighborhood. In such scenarios the intended recipients are identified based on their locations rather than their individual identities.

6.6.1 Routing objective

As we explained in Chapter 4, applications and routing algorithms that are developed for mobile DTNs have a better chance of wide deployment by mobile users, if they have a more efficient utilization of the communication resources. In mobile DTN scenarios mobile users rely on other mobile users to transmit their data to the intended locations or recipients. Carrying other users' data is costly for data carriers in terms of communication resources. The fact that these data carriers should spend their communication resources for the benefits of others is a burden on practical deployment of mobile DTN routing protocols. The most valuable communication resource is energy. Therefore, a reasonable evaluation metric for the routing algorithms is how much energy they spend to achieve their routing goals.

Epidemic routing is proven to provide the highest delivery rate and the lowest delay in transmitting data to its destination [Vahdat and Becker, 2000] in mobile DTN scenarios. Epidemic routing, however, has the least efficient usage of the communication resources, especially energy, as it floods all mobile users with all the data. On the other side of the

energy spectrum, however, we could consider an algorithm in which the sender does not transmit the data to any other mobile user. Therefore, data could reach the destination only if sender arrives at that destination. This algorithm has the least energy consumption but naturally the longest expected transmission delay. Considering these two extreme cases, we can define scenarios in between where some percentage of wireless contacts is excluded from any data exchange. The evaluation objective of our Markov routing algorithm is to investigate where our algorithm is located in this performance spectrum. In other words, we investigate the trade-off between energy and delay by our algorithm or the amount of delay per unit of consumed energy.

6.6.2 Algorithm design

The Markov chains we extract from users' daily movement traces explain the daily movement profile of each user and the daily rates by which they transition from one location to another. Using these rates, we can calculate the expected number of steps or transitions until each Markov chain arrives at a specific state. Since the time between each two transitions is fixed we can map the number of transitions to the expected amount of time that is needed until a user reaches a certain destination after starting at some other location. This expected time which is called *absorption time* is a determining factor in our routing algorithm. By comparing the absorption times of the mobile users to a certain destination, the source of a data object is capable of selecting the best possible carrier for that data. Because of the importance of the absorption times in our routing algorithm, next we briefly explain how to calculate these values for a general Markov chain.

6.6.2.1 Calculating the Absorption Time

If we represent the Markov chain with n number of states as $(X_n)_{n \geq 0}$ and transition matrix P then the *absorption time* at a subset A of these states is the random variable $H^A : \Omega \rightarrow \{0, 1, 2, \dots\}$ given by:

$$H^A(\omega) = \inf\{n \geq 0 : X_n(\omega) \in A\}$$

Then, the probability of, starting from state i , the Markov chain $(X_n)_{n \geq 0}$ ever ending at A is

$$h_i^A = P_i(H^A < \infty).$$

Therefore, the mean time taken for $(X_n)_{n \geq 0}$ to reach A is given by

$$K_i^A = E_i(H^A) = \sum_{n < \infty} nP(H^A = n) + \infty P(H^A = \infty).$$

We can write this equivalently as:

$$H_i^A = P_i(\text{hit } A),$$

$$K_i^A = E_i(\text{time to hit } A).$$

These quantities can be calculated explicitly by means of linear equations associated with the transition matrix, P . These systems of linear equations solve for the vector of mean absorption times, k^A , and the vector of mean absorption probabilities, h^A . The system of linear equations to solve for the absorption times is a set of recursive equations that allows us to solve for the mean times that the chain ends up at certain states.

$$\begin{cases} k_i^A = 0, & \text{for } i \in A \\ k_i^A = 1 + \sum_{j \notin A} p_{ij} K_j^A, & \text{for } i \notin A \end{cases}$$

6.6.2.2 Algorithm Mechanics

As we explained previously, we represent users' daily movement models as bigram tables or transition matrices. Our algorithm's decision-making process for selecting data carriers is based on the absorption times of the mobile users to the destination. The algorithm calculates each user's mean absorption times from each location to another by solving a system of linear equations, as discussed in Section 6.6.2.1. In a format similar to bigram tables, these absorption times also could be represented as matrices of size $n \times n$, where n is the number of locations or states of the Markov chain. Each element, T_{ij} , of these matrices at row i and column j shows the mean absorption time from location i to location

j for a specific daily activity of a certain user. All users have the knowledge of their own absorption time tables.

When some users meet at some location i , there are two subsets of users. One subset of users has some data that needs to be transmitted to some destination location, j . We call this subset the source, and we represent it with s . The second subset is users who could be potential forwarders of the, data because they could reach location j with some probability. We call this second subset of users relays and we symbolize the subset with r .

Users in the relay subset notify the users in the source subset of their mean absorption times to the destination location j or T_{ij}^r . Users in the source subset compare these values with their own absorption times to destination j or T_{ij}^s . After these comparisons source decides to transmit the data to the relay if:

$$T_{ij}^r \leq w_j T_{ij}^s,$$

where w_j is a positive value that weighs the importance of the destination location, j . The default value for w_j is one and the larger this value, the more critical the destination location, and the less selectively the algorithm acts to choose the potential relay. By critical destination location we mean data needs to get there as fast as possible and therefore more transmissions should occur. In other words, the larger w_j implies a wider filter to choose the relay because choosing more relays means increasing the chance of data delivery to the critical destination.

6.6.3 Evaluation of the algorithm by Monte Carlo simulations

We have evaluated the performance of our Markov based routing algorithm by Monte Carlo simulations. In order to have a realistic evaluation of this algorithm, we have implemented a mobility generator module that generates mobility of the mobile users based on the real-world traces. The input to this mobility module is the transition matrices of the Markov chains that are learned from the real-world traces, as explained in Section 6.5. In addition to these realistic transition matrices, artificial users could also be generated by supplying the module with randomly generated transition matrices. This mobility generator module uses Monte Carlo methods to generate mobility of the users in the simulation scenario by

sampling from these supplied transition matrices. Monte Carlo simulations are a class of computational simulations that rely on repeated random sampling to compute their results [Rubinstein and Kroese,], [Raychaudhuri, 2008].

6.6.3.1 Simulation Setup

Our simulation scenarios consist of 50 mobile users who move between 100 locations. For each user we choose 10 mostly visited locations based on their GPS traces. As we explained before, we ignore the locations in which users have very few pings as they do not have any significance in our final results. We regenerate each user's mobility based on the transition matrices that we have extracted from Sense Network's traces. These transitions matrices define the inputs to the mobility generator engine.

Based on the Monte Carlo guidelines [Rubinstein and Kroese,], we generate a random movement sample as the input, feed this input to a deterministic routing algorithm simulator engine, and then evaluate the output. In order to generate the movements, we sample from each user's corresponding transition matrix to emulate movements of the user from one location to another for each simulation scenario. Therefore, each simulation round in the scenario can be interpreted as the current state of the Markov chain or as a user's current location. The current states of all the users are supplied to the routing algorithm simulator engine as an input. Based on this input, if some users meet at some location, the simulator engine emulates the behavior of the routing algorithm, as explained in Section 6.6.2.2. The simulation scenario continues until the data object reaches its destination location. After the simulation ends the amount of the transmission delay and number of total transitions are recorded.

At each simulation scenario one of the users is uniformly selected as the source of the data object which needs to be transmitted to an also randomly selected destination location. The output of each simulation scenario is the delay until the data arrives at its destination and the total amount of energy that is spent for this data transmission. Our metric to measure the delay is the time unit between two consecutive GPS pings. Based on our model, every transition from one state to another in the Markov chains happens at these time units. In our simulation scenarios, the metric to represent the amount of consumed

energy is the total number of transmissions that takes place to transmit a data object to its destination. We assume the amount of energy to transmit a single data object per wireless contact is constant across all mobile devices. At the end the results of the 100 generated simulation scenarios are aggregated to decrease the impact of the randomness in the final averaged results.

6.6.3.2 Evaluation of Results

Since epidemic routing produces the least amount of transmission delay and highest amount of Energy consumption, our main comparison base is epidemic. We define a performance spectrum that consists of epidemic and other routing algorithms which are similar to epidemic but deliberately exclude a percentage of wireless contacts from data transmission. Therefore, we define a family of epidemic algorithms and we call them α -epidemic, where *alpha* is the percentage of the wireless contacts participating in the data exchange. When α is 1 we have a fully epidemic algorithm. The less the amount of α the more energy-aware is the algorithm but it produces larger amounts of transmission delay. Our simulation goal is to investigate where our Markov based algorithm falls in this performance spectrum compared to these algorithms with different values of *alpha*. In other words, if the class of α -epidemic algorithms has a total performance curve in the form of $\text{delay} = f(\text{energy})$, the Markov based algorithm is more energy efficient if its performance curve lies beneath the class of α -epidemic's, meaning $\text{delay} < f(\text{energy})$. In other words, an algorithm has a better performance if it produces less delay per unit of consumed energy.

In our first series of simulations, we create different scenarios for epidemic and α -epidemic routing with different values for α . Then, we compare the performance of our Markov based routing algorithm with this family of epidemic and α -epidemic algorithms. The goal of simulating different forms of epidemic is to generate a testbed to evaluate the performance curve of the Markov based routing compared to $\text{delay} = f(\text{energy})$ of the family of α -epidemics. In the first epidemic scenario, senders transfer the data object to the 100% of the mobile users they encounter. In the rest of the α -epidemic scenarios senders transfer data to a randomly selected subset of mobile encounters. The size of this subset varies from 50% to 10% of the entire encounters across different simulations. In all these

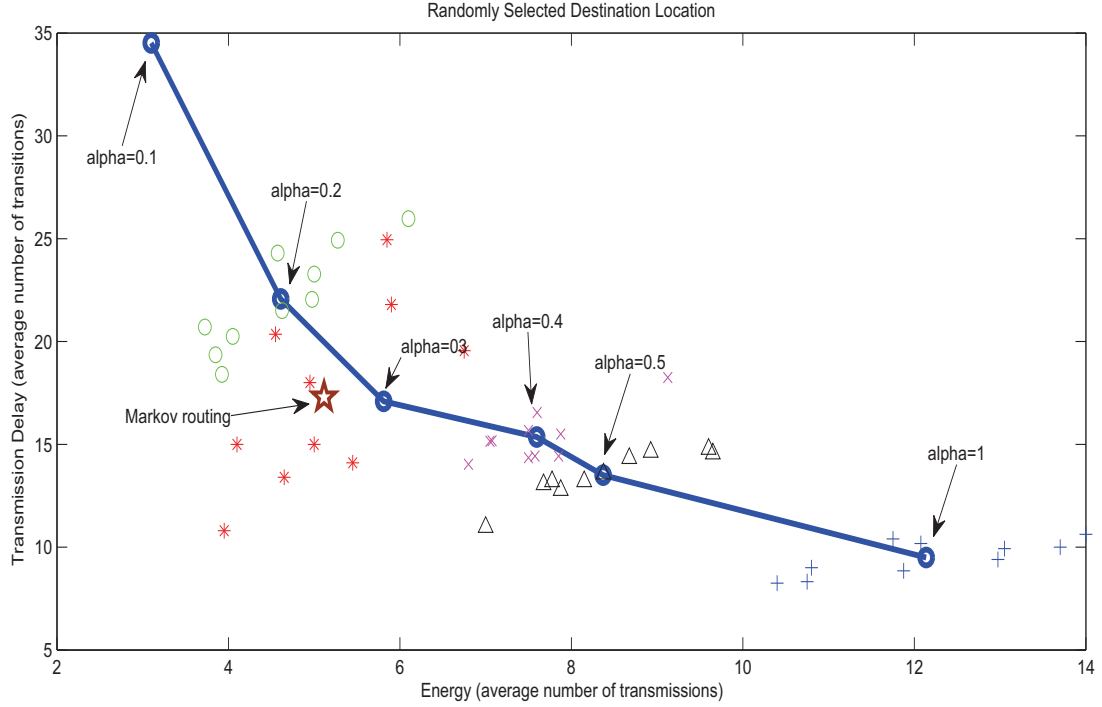


Figure 6.10: Transmission delay vs. consumed energy for a randomly selected destination.

simulation scenarios at each round of the simulation the destination location is selected randomly from the 100 total locations. The simulation results are represented in Figure 6.10. We change α from 1, meaning fully epidemic, to zero, meaning the sender does not transmit data over any wireless contact, and the data object reaches the destination when the sender itself arrives at the destination. The abscissa measures the consumed energy in terms of the average number of transmissions until the data object reaches its destination while the y -axis measures the average delay of these transmissions in terms of elapsed time units. The solid curve is calculated from the average values of delay and energy produced by each method of the α -epidemic family. In other words, this curve represents the $\text{delay} = f(\text{energy})$ relationship for the α -epidemic family. The point on the figure that is highlighted with an arrow is the average value of the delay and consumed energy that resulted from the Markov based routing algorithm. As we can observe in this figure, selecting the relays based on their absorption times to random destinations does not gain much over the performance much because both resulted delay and energy values are very close to the

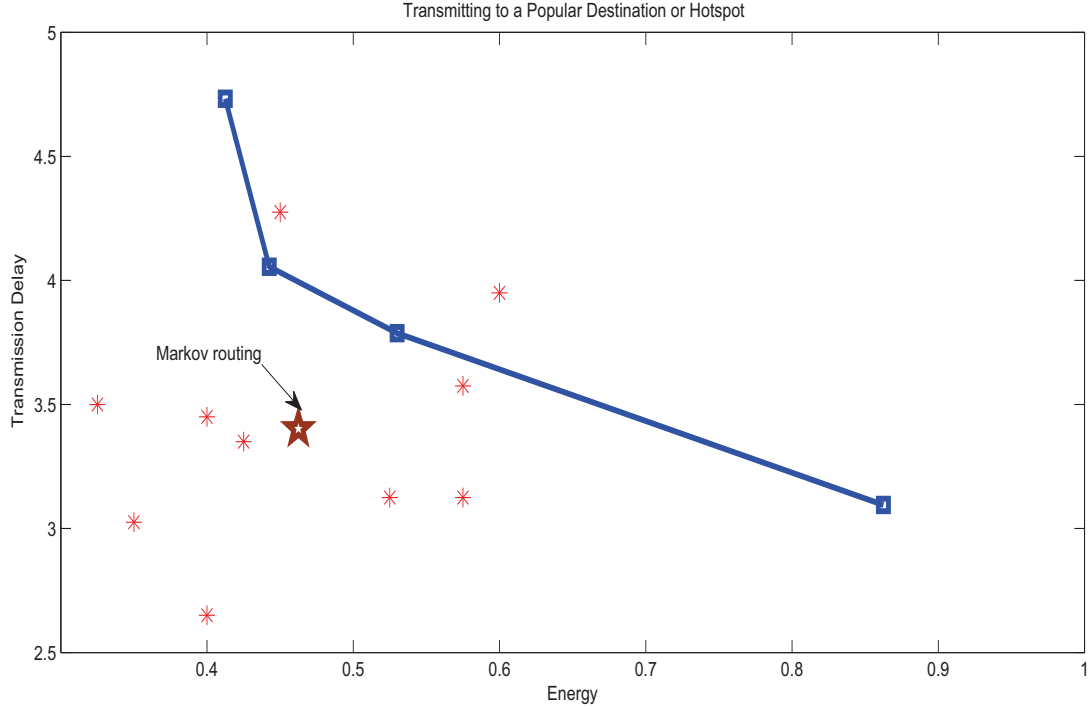


Figure 6.11: Transmission delay vs. consumed energy for a popular destination or hotspot.

α -epidemic where $\alpha = 0.3$ or when the α -epidemic algorithm which randomly selects 30% of the encountered mobile users as potential relays. Therefore, using the Markov based algorithm does not improve the result much over using α -epidemic with $\alpha = 0.3$ which does not use any knowledge of the mobility of the mobile users.

After further investigations in the movement trace database, we realized that all users have large mean absorption times to the majority of the locations, because the majority of the locations are not visited often by many mobile users. For example, some user's home location in the suburbs of San Francisco is hardly ever visited by any of the other users. Having large absorption times to many locations means our algorithm cannot really take advantage of the absorption times in its routing decisions.

In order to observe the impact of the absorption times in the routing decisions, we need to select our destinations from the popular locations. Popular locations or *hotspots* are locations or neighborhoods that are visited more often by a larger population of users, for example, downtown San Francisco. Since these locations are popular, naturally more users

have less mean absorption times to these locations. Figure 6.11 illustrates the results of this series of simulations for the class of α -epidemic and Markov based algorithms. As we observe in the figure, all algorithms show improved results in terms of consuming less energy and achieving less delay. This change of trend happens, because, since the destination is a hotspot, there is a higher chance that it is visited more frequently by most of the users. But more important, there is a significant improvement in the Markov based routing compared to the α -epidemic family. The solid line in the figure again shows the performance of the family of α -epidemic. The dots on the figure are the simulation results of our Markov based routing algorithm with the average point emphasized with an arrow. As we observe in this figure, for almost all simulation rounds of the Markov based algorithm we have $\text{delay} < f(\text{energy})$ compared to the epidemic. Moreover, the average point, or center of gravity of the Markov based algorithm falls considerably below the epidemic's curve. This confirms that for the same amount of delay the Markov based algorithm consumes 32% less energy. Also, its energy consumption is similar to the energy consumption of an α -epidemic algorithm, which randomly drops 70% of the wireless contacts, while the delay of the Markov based algorithm is as low as the delay of an α -epidemic which drops 25% of the contacts. Our Markov based routing algorithm is more suitable for transmitting data to the popular neighborhoods or hotspots in a more energy efficient manner without increasing the transmission delay.

6.7 Conclusion

We studied different aspects of human mobility and investigated how the knowledge of this mobility could affect the performance of the routing algorithms for mobile DTNs. For our mobility behavior studies we used Sense Networks's GPS traces, which are collected from a large population of mobile users. Using these traces, we identified that latent factors such as time and type of a day could impact mobile users' mobility patterns. We used a class of statistical models called n-grams to extract meaningful patterns for human movement. We also evaluated the accuracy of unigram, bigram, and trigram models in describing mobile users' movements. Bigram and trigram models produce less error compared to

the unigram models. Furthermore, we realized that bigram modeling is a suitable enough candidate to explain users' movements and to predict their future locations. Extracting and evaluating the bigram tables led us to use a second-order Markov model, or Markov chains as a promising pattern recognition tool to study human mobility.

We used these extracted Markov chains from users' movement traces and designed a new routing algorithm based on users' absorption times to different locations. In our routing algorithm, the sender of the data prefers mobile users who have lower absorption times to the destination as the potential relays to carry that data object. After evaluations of this Markov based routing algorithm, we showed by simulations that this routing algorithm consumes considerably less energy to carry data objects to the popular destinations than the family of epidemic routing. Furthermore, our routing algorithm achieves this superior performance without compromising much on transmission delay. Our algorithm's performance in terms of transmission delay is similar to that of an α -epidemic routing, which randomly drops 25% of the wireless contacts. However, our algorithm's energy consumption is as low as that of the randomized epidemic routing, which doesn't randomly transmit to 70% of the wireless contacts. Therefore, our algorithm offers an acceptable trade-off between energy and transmission delay ,especially for the applications which are more energy sensitive but not very time critical.

Table 6.1: Comparison of the movement traces that have been used in trace-driven studies

Project	Cambridge	Hong Kong	Infocomm	Toronto	UCSD	Dartmouth	MIT BT	MIT GSM	Nature GSM (D1,D2)
Device	iMote	iMote	iMote	PDA	PDA	Laptop/PDA	Cellphone	Cellphone	Cellphone
Network Type	Bluetooth	Bluetooth	Bluetooth	Bluetooth	WiFi	WiFi	Bluetooth	GSM	GSM
Contact Type	direct	direct	direct	direct	AP-Based	AP-Based	direct	AP-Based	AP-Based
Duration (Days)	3	5	4	16	77	114	246	246	D1 - 180 D2 - 7
Time Granularity (seconds)	120	120	120	120	120	300	300	10	D1: a call or txt D2: 7200
# Devices Participating	12	37	41	23	273	6,648	100	100	D1: 100,000 D2: 206
# Internal Contacts	4,229	280	22,459	2,802	195,364	4,058,284	54,667	572,192	Not Available

Part IV

Conclusions

Chapter 7

Conclusions

Modern wireless networks span across different connection technologies and can be constructed with or without any infrastructure. Moreover, today mobile devices are equipped with different wireless technologies such as GSM radio, IEEE 802.11, Bluetooth and other interfaces. With this variety in networking environments and networking interfaces, communication opportunities could become possible at any time and any place, even where no networking infrastructure is available. Because mobile users migrate between globally connected environments to local standalone mobile networks, they experience disruption in their communication services. Although the global communication might become disrupted, mobile users might find new connection opportunities with other mobile users in standalone mobile networks. We classified this type of networking as mobile Disruption-Tolerant Networking (DTN). We discussed open research problems in mobile DTNs from three different viewpoints; applications, routing algorithms, and mobility behavior of mobile users and how the knowledge of the mobility can improve the performance of the routing algorithms and applications.

In the application layer we developed 7DS as a platform that provides different types of mobile disruption-tolerant applications. Furthermore, 7DS provides a modular software platform that serves as an environment to develop mobile disruption-tolerant applications. Modular design of 7DS facilitates future expansion of the platform independently from the underlying core system. In Chapter 3 we introduced a new class of applications and specified their fundamental requirements to operate in disruption-tolerant environments. The main

significance of these applications is in covering a broad range of core functionalities. Considering these primary functionalities, we designed and implemented 7DS's primary APIs for device and resource discovery, service resolution, zero configuration network setup, store-carry-forward routing, the search engine, messaging and delta encoding.

We further implemented four disruption-tolerant applications for mobile DTNs using the modules and APIs provided by the 7DS platform. The 7DS platform is able to provide disruption-tolerant versions of two core Internet services, namely web access for information retrieval and email for delivering messages from mobile nodes to the Internet. 7DS also offers peer-to-peer communication among mobile users by its file synchronization applications and communication with stationary data centers through its bulletin board system (BBS). Data sharing components of the 7DS system offer both push-based and pull-based data sharing models through BBS and file synchronization applications respectively. In the pull-based file synchronization model, mobile applications are able to download new files from newly discovered mobile peers automatically. These uploads and downloads are transparent to the application. In the push-based BBS, however, information is shared based on the application's preferences. Setting up 7DS and running its applications on any device or computer is fairly easy. After the initial setup, 7DS-enabled devices can automatically exchange information with other mobile peers to overcome the lack of the global connectivity. When a global communication becomes available, for example through a WiFi hotspot, 7DS-enabled devices switch automatically to the new communication mode with the global Internet. We have released the 7DS platform and its applications for public's use on SourceForge [Moghadam *et al.*, 2006].

The second problem scope that we addressed in this thesis was the problem of routing in mobile DTNs. Routing in mobile DTNs, specifically, is much more challenging than routing in the traditional connected networks such as the Internet. All previously introduced routing methodologies for the connected networks even mobile ad-hoc networking protocols fail to operate in mobile DTNs due to the lack of any infrastructure and frequently changing topology. Furthermore, in the routing scope, most of the algorithms that were previously designed for the mobile DTNs were focused on the one-to-one routing paradigm between a source and a specific destination and the problem of multicasting was not addressed prop-

erly. We introduced interest-aware algorithms as intelligent content distribution algorithms for communication with groups of mobile users based on their interests. These algorithms are able to learn users' behaviors and their interests over time. Our interest-aware algorithms, unlike the classical multicast, do not need any knowledge of the group memberships of mobile users and the topology of the network to properly multicast the data. Interest-aware algorithms are able to classify mobile users based on their interests and automatically distribute data among communities of interest. The interest-aware algorithms help the participating users in the routing, which are called relays, to also locate and extract the data content they are interested in. This gives mobile users a stronger incentive to share their communication resources to carry other users' data as they also benefit from it. After analyzing the performance of the interest-aware algorithms and their insufficiencies, we evolved the interest-aware communication model toward a more comprehensive model which we named PEEP. PEEP solves the interest-aware's greediness and energy consumption issues. We solved the energy issue by introducing transmit budget which is a data transmission token to determine the amount of data that is allowed to be transferred over each wireless contact. PEEP also solves the interest-aware's greediness by estimating the global popularity of the data items from the history of the interests of the mobile users who have been encountered. PEEP, then, includes the globally popular items as well as the locally interesting ones in each data transmission. We implemented the PEEP protocol in the ONE simulator and evaluated its performance in terms of speed of the distribution of items of interests. Our simulation results show that our protocol, compared to the epidemic routing, produces a faster mechanism for diffusion of useful items throughout the network. The PEEP algorithm with the popularity estimation method distributes items of interest among mobile users on average 44% faster than epidemic. Furthermore, our simulations verify that PEEP's performance is robust and does not depend on the choice of the underlying mobility model.

After discussing the data distribution problem in mobile DTNs we focused on mobility models for mobile DTNs. We discussed that store-carry-forward communication which is the heart of all routing algorithms for mobile DTNs exploits nodes' mobility to bring messages closer to their destinations. Therefore, mobility impacts the conditions in which

routing protocols must operate and types of applications that could be provided for mobile DTNs. The mobility models are designed to describe the movement pattern of mobile users, and how their location, velocity and acceleration change over time. Since mobility patterns play a significant role in determining the performance of mobile protocols and applications, it is necessary for mobility models to emulate the movement pattern of targeted real life applications in a reasonable way. Otherwise, the observations made and the conclusions drawn from the simulation studies may be misleading. Despite the importance of the knowledge of the mobility in designing more practical routing algorithms and applications for mobile DTNs, there are still many unresolved issues in this area of the research. In Chapter 6 we reviewed different types of mobility models that have been defined to represent the mobility of mobile users. There are two main classes of mobility models that are derived for mobile DTNs; trace-based and synthetic mobility models. One intuitive method to create realistic mobility patterns would be to construct trace-based mobility models, in which accurate information about the mobility traces of users could be provided. Synthetic mobility models are popular as they are easy to use and more general simulation scenarios with arbitrary numbers of mobile users can be produced by these models. However, most widely used synthetic models are very simplistic, because their main focus is their ease of implementation. Additionally, these synthetic models are generated based on random parameters which differ significantly from reality. Trace-based mobility models, on the other hand, are extracted from movement traces of real-world mobile users. However, there are a very limited number of available real-world traces in the research domain. Also, these traces mostly have been collected from specific movement scenarios and specific populations of mobile users.

We used a new set of movement traces to analyze spatial and temporal aspects of mobile users' movement. For some trace-driven mobility models, the movement of a mobile node is likely to be affected by its movement history. We refer to this type of mobility model as mobility model with temporal dependency. In other mobility scenarios, the mobile nodes tend to travel in a correlated manner. We refer to such models as mobility models with spatial dependency. We used Sense Networks' GPS traces that are collected from a large population of mobile users for to study spatial and temporal aspects of mobile users' move-

ments. Using these traces, we studied how underlying factors such as the history of visited locations and the day and time of the travels could impact mobile users' mobility patterns. We derived a Markov routing algorithm for mobile DTNs based on spatio-temporal modeling of human movement Data. We used a class of statistical models called n-grams to extract meaningful patterns for human movement. We evaluated the accuracy of unigram, bigram and trigram models in describing mobile users' movements. Bigram models along with trigrams produce less error compared to the unigram models in calculating the likelihoods of the users' future locations. We realized that bigram modeling is a suitable enough candidate to explain users' movements and to predict their future locations. Extracting and evaluating the bigram tables led us to use a second-order Markov model or Markov chains as a promising pattern recognition tool to study human mobility.

We used these extracted Markov chains from users' movement traces and designed a new routing algorithm based on users' absorption times to different locations. Absorption times are mean expected times until a user reaches a specific destination after starting at some arbitrary location. In the relay selection process of our routing algorithm, the sender of the data prefers mobile users who have less absorption times to the destination as the carriers of that data. We implemented this Markov-based routing algorithm for mobile DTNs and evaluated its performance by simulations. In our simulations, the movement of the mobile users is generated based on the Markov chains that we extracted from the Sense Networks' traces. The objective of this evaluation is to measure the total amount of consumed energy to carry the data object to its destination location for a specific amount of transmission delay.

We considered epidemic routing as the base of comparison to evaluate the performance of our Markov-based routing algorithm. We compared the performance results to epidemic as epidemic routing is proven to produce the least amount of delay in data propagation. Epidemic routing, on the other hand, has been proven to have the most inefficient usage of communication resources, most importantly energy. In order to have a meaningful scale to measure the trade-offs between energy and delay, we generated different scenarios for the epidemic routing in our simulations. As well as simulating a fully epidemic scenario in which senders of the data transmit the data to all mobile nodes that they encounter,

we also simulated a family of α -epidemic scenarios. In these α -epidemic scenarios, senders of the data transmit the data to a randomly selected subset of mobile users. Naturally, the larger this subset, the closer the performance of the α -epidemic to the fully epidemic scenario, in terms of consumed energy and produced delay. The final goal of our simulations was to determine where the performance of our Markov-based routing algorithm is situated compared to the family of the α -epidemics. After the evaluations, we proved that our routing algorithm consumes considerably less energy to carry data objects to the popular destinations than the family of α -epidemic routing. Furthermore, our routing algorithm achieves this superior performance without compromising much on transmission delay. The transmission delay of our Markov-based algorithm is close to the transmission delay of an α -epidemic routing which transmits data to 75% of the wireless contacts. However, our algorithm's energy consumption is as less as α -epidemic routing which transmits to a randomly selected subset of 40% of the wireless contacts. Therefore, our algorithm offers an acceptable trade-off between energy and transmission delay especially for the applications which are more energy sensitive but not very time critical.

Part V

Appendices

Appendix A

Calculating Absorption Probabilities and Absorption Times for Markov Chains

In this appendix we explain the general mechanism and theorems to calculate absorption times and absorption probabilities of a Markov chain with examples. For the proof of theorems we refer the reader to [Norris, 1998] and [Karlin and Taylor, 1981] where the examples in this appendix are from.

If we represent the Markov chain with n number of states as $(X_n)_{n \geq 0}$ and transition matrix P then the *absorption time* at a subset A of these states is the random variable $H^A : \Omega \rightarrow \{0, 1, 2, \dots\} \cup \{\infty\}$ given by [Norris, 1998]:

$$H^A(\omega) = \inf\{n \geq 0 : X_n(\omega) \in A\}$$

where we agree the infimum of the empty set \emptyset is ∞ . The probability of starting from state i the Markov chain $(X_n)_{n \geq 0}$ ever ends at A is:

$$h_i^A = P_i(H^A < \infty).$$

when A is a closed class, h_i^A is called *absorption probability*. The mean time taken for $(X_n)_{n \geq 0}$ to reach A is given by:

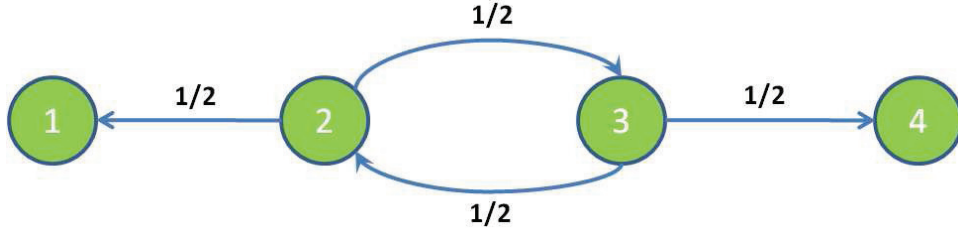


Figure A.1: Markov chain of example A.0.1.

$$K_i^A = E_i(H^A) = \sum_{n < \infty} nP(H^A = n) + \infty P(H^A = \infty).$$

We can write equivalently as:

$$H_i^A = P_i(\text{hit } A), \quad K_i^A = E_i(\text{time to hit } A).$$

These quantities can be calculated explicitly by means of linear equations associated with the transition matrix P . Before we give the general method, here is a simple example.

A.0.1 Example

Consider the Markov chain with the diagram shown in Figure A.1.

Starting from state 2, we calculate the probability of absorption in state 4. We also calculate the time until the chain is absorbed in state 1 or state 4.

$$h_i = P_i(\text{hit } 4), \quad k_i = E_i(\text{time to hit } \{1, 4\}).$$

It is obvious from the definition that $h_1 = 0$, $h_4 = 1$ and $k_1 = k_4 = 0$. Suppose now we start at 2, and consider the situation after making one step. With probability $1/2$ we jump to 1 and with probability $1/2$ we jump to 3. So we have:

$$\begin{aligned} h_2 &= \frac{1}{2}h_1 + \frac{1}{2}h_3, \\ k_2 &= 1 + \frac{1}{2}k_1 + \frac{1}{2}k_3. \end{aligned}$$

The 1 appears in the second formula because we count the time for the first step. Similarly, for the state 3 we have:

$$\begin{aligned} h_3 &= \frac{1}{2}h_2 + \frac{1}{2}h_4, \\ k_3 &= 1 + \frac{1}{2}k_2 + \frac{1}{2}k_4. \end{aligned}$$

Therefore, after solving the above system we get:

$$\begin{aligned} h_2 &= \frac{1}{2}h_3 = \frac{1}{2}\left(\frac{1}{2}h_2 + \frac{1}{2}\right), \\ k_2 &= 1 + \frac{1}{2}k_3 = 1 + \frac{1}{2}\left(1 + \frac{1}{2}k_2\right). \end{aligned}$$

So, starting from state 2, the probability of hitting state 4 is $1/3$ and the mean time to absorption is 2. Note that in writing down the first equation for h_2 and k_2 we made implicit use of the Markov property, in assuming that the chain begins afresh from its new position after the first jump. Here is a general result for hitting probabilities.

Theorem A.0.1 *The vector of hitting probabilities $h^A = (h_i^A : i \in I)$ is the minimal non-negative solution to the system of linear equations*

$$\begin{cases} h_i^A = 1, & \text{for } i \in A \\ h_i^A = \sum_{j \in I} p_{ij} h_j^A, & \text{for } i \notin A \end{cases} \quad (\text{A.1})$$

(Minimality means that if $x = (x_i : i \in I)$ is another solution with $x_i \geq 0$ for all i , then $x_i \geq h_i$ for all i .)

Going back to the example A.0.1 and applying theorem A.0.1, the system of linear equations A.1 for $h = h^{\{4\}}$ are given here by:

$$\begin{aligned} h_4 &= 1, \\ h_2 &= \frac{1}{2}h_1 + \frac{1}{2}h_3, \\ h_3 &= \frac{1}{2}h_2 + \frac{1}{2}h_4 \end{aligned}$$

so that

$$h_2 = \frac{1}{2}h_1 + \frac{1}{2}\left(\frac{1}{2}h_2 + \frac{1}{2}\right)$$

and

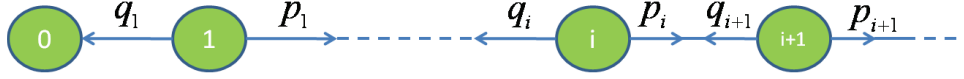


Figure A.2: Birth-and-death-chain

$$\begin{aligned} h_2 &= \frac{1}{3} + \frac{2}{3}h_1, \\ h_3 &= \frac{2}{3} + \frac{1}{3}h_1. \end{aligned}$$

The value of h_1 is not determined by the system A.1, but the minimality condition now makes us take $h_1 = 0$, so we recover $h_2 = \frac{1}{3}$ as before. Of course, the extra boundary condition $h_1 = 0$ was obvious from the beginning so we built it into our system of equations and did not have to worry about minimal non-negative solutions.

A.0.2 Example (Birth-and-death chain)

Consider the Markov chain illustrated in Figure A.2.

Where, for $i = 1, 2, \dots$, we have $0 < p_i = 1 - q_i < 1$. Here state 0 is the absorbing state and we wish to calculate the absorption probability starting from state i .

Such a chain may serve as a model for the size of a population, recorded each time it changes, p_i being the probability that we get a birth before a death in a population of size i . Then $h_i = P(\text{hit } 0)$ is the extinction probability starting from i .

We write down the usual system of equations

$$\begin{aligned} h_0 &= 1, \\ h_i &= p_i h_{i+1} + q_i h_{i-1}, \text{ for } i = 1, 2, \dots \end{aligned}$$

This recurrence relation has variable coefficients so the usual technique fails. But we consider that if $u_i = h_{i-1} - h_i$, then $p_i u_{i+1} = q_i u_i$, so

$$u_{i+1} = \left(\frac{q_i}{p_i} \right) u_i = \left(\frac{q_i q_{i-1} \dots q_1}{p_i p_{i-1} \dots p_1} \right) u_1 = \gamma_i u_1$$

where the final equality defines γ_i . Then

$$u_1 + \dots + u_i = h_0 - h_i$$

so

$$h_i = 1 - A(\gamma_0 + \dots + \gamma_{i-1})$$

where $A = u_1$ and $\gamma_0 = 1$. At this point A remains to be determined. In the case $\sum_{i=0}^{\infty} \gamma_i = \infty$, the restriction $0 \leq h_i \leq 1$ forces $A = 0$ and $h_i = 1$ for all i . But if $\sum_{i=0}^{\infty} \gamma_i < \infty$ then we can take $A > 0$ so long as

$$1 - A(\gamma_0 + \dots + \gamma_{i-1}) \geq 0 \text{ for all } i.$$

Thus the minimal non-negative solution occurs when $A = (\sum_{i=0}^{\infty} \gamma_i)^{-1}$ and then

$$h_i = \frac{\sum_{j=i}^{\infty} \gamma_j}{\sum_{j=0}^{\infty} \gamma_j}$$

In this case, for $i = 1, 2, \dots$, we have $h_i < 1$, so the population survives with positive probability.

Now, we represent the general result on mean hitting times. Recall that $k_i^A = E_i(H^A)$, where H^A is the first time the chain $(X_n)_{n \geq 0}$ hits A . We use the notation 1_B for the indicator function of B , so, for example, $1_{X_1=j}$ is the random variable equal to 1 if $X_1 = j$ and equal to 0 otherwise.

Theorem A.0.2 *The vector of mean hitting times $k^A = (k_i^A : i \in I)$ is the minimal non-negative solution to the system of linear equations*

$$\begin{cases} k_i^A = 0, & \text{for } i \in A \\ k_i^A = 1 + \sum_{j \notin A} p_{ij} k_j^A, & \text{for } i \notin A \end{cases} \quad (\text{A.2})$$

Based on this system of linear equations, calculating the mean absorption times for a Markov chain is very similar to calculating the absorption probabilities. Therefore, here we don't provide any example regarding the absorption times calculations.

Part VI

Bibliography

Bibliography

- [Abdulla and Simon, 2007] M. Abdulla and R. Simon. The impact of the mobility model on delay tolerant networking performance analysis. *40th Annual Simulation Symposium, 2007. ANSS '07.*, pages 177 – 184, March 2007.
- [Adamic and Huberman, 2002] Lada Adamic and Bernardo Huberman. Zipfs law and the internet. *Glottometrics*, 3:143–150, 2002.
- [Arezu Moghadam and Henning Schulzrinne, 2009] Arezu Moghadam and Henning Schulzrinne. Interest-aware content distribution protocol for mobile disruption-tolerant networks. In *10th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, Kos, Greece*, June 2009.
- [Balani, 2007] Rahul Balani. Energy Consumption Analysis for Bluetooth, WiFi and Cellular Networks. Network and Embedded System Technical Report TR-UCLA-NESL-200712-01, University of California in L.A., UCLA, California, December 2007. Available at <http://nesl.ee.ucla.edu/document/show/254>.
- [Berry *et al.*, 1995] Michael W. Berry, Susan T. Dumais, and Gavin W. O’Brien. Using linear algebra for intelligent information retrieval. *SIAM Rev.*, 37(4):573–595, 1995.
- [BitTorrent, 2001] BitTorrent. BitTorrent protocol, 2001. <http://wiki.theory.org/BitTorrentSpecification>.
- [Boldrini *et al.*, 2007] C. Boldrini, M. Conti, J. Jacopini, and A. Passarella. HiBOp: a history based routing protocol for opportunistic networks. *IEEE International Sympo-*

sium on a World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007, Helsinki, Finland., June 2007.

[Bonjour, 2002] Bonjour. Apple Computer's Bonjour, 2002. <http://developer.apple.com/networking/bonjour/>.

[Boudec and Vojnovic, 2006] Jean-Yves Le Boudec and Milan Vojnovic. The random trip model: stability, stationary regime, and perfect simulation. *IEEE/ACM Trans. Netw.*, 14(6):1153–1166, 2006.

[Brockmann and Theis, 2008] Dirk Brockmann and Fabian Theis. Money circulation, trackable items, and the emergence of universal human mobility patterns. *Pervasive Computing, IEEE*, 7(4):28–35, Oct.-Dec. 2008.

[Brown *et al.*, 2001] Michael K. Brown, Andreas Kellner, and Dave Raggett. Stochastic language models (N-Gram) specification. W3C Working Draft 3, January 2001. <http://www.w3.org/TR/ngram-spec/>.

[Burgess *et al.*, 2006] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, April 2006.

[Camp *et al.*, 2002] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing (WCMC): Special Issue On Mobile Adhoc Networking: Research, Trends And Applications*, 2:483–502, 2002.

[Cerf *et al.*, 2007] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. RFC 4838: Delay-tolerant networking architecture, April 2007. <http://www.ietf.org/rfc/rfc4838.txt>.

[Chaintreau *et al.*, 2007] A. Chaintreau, P. Hui, C. Diot, R. Gass, and J. Scott. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Transactions on Mobile Computing*, 6(6):606 – 620, 2007.

- [Chen *et al.*, 2006] Yang Chen, Jeonghwa Yang, Wenrui Zhao, M. Ammar, and E. Zegura. Multicasting in sparse MANETs using message ferrying. *Wireless Communications and Networking Conference, 2006. WCNC 2006. Las Vegas, NV.*, 2:691–696, 2006.
- [Chiang, 1998] Ching-Chuan Chiang. *Wireless network multicasting*. PhD thesis, 1998.
- [Citysense, 2008] Citysense. Citysense, 2008. <http://www.citysense.com/>.
- [Clausen and Jacquet, 2003] T. Clausen and P. Jacquet. RFC 3626: Optimized Link State Routing Protocol (OLSR), October 2003. <http://www.ietf.org/rfc/rfc3626.txt>.
- [Conti *et al.*, 2007] M. Conti, F. Delmastro, and A. Passarella. Context-aware file sharing for opportunistic networks. *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE International Conference on*, Oct. 2007.
- [CVS, 1990] CVS. CVS - concurrent versions system, 1990. <http://www.nongnu.org/cvs/>.
- [Daly and Haahr, 2007] E. M. Daly and M. Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *MobiHoc '07: Proceedings of the 8th ACM international symposium on mobile ad hoc networking and computing*, pages 32–40, New York, NY, USA, 2007. ACM.
- [Davies, 2000] V. Davies. Evaluating mobility models within an ad hoc network. Master’s thesis, 2000.
- [Droms, 1997] R. Droms. DHCP - Dynamic Host Configuration Protocol. In *RFC 2131.*, March 1997.
- [Dropbox, 2008] Dropbox. Dropbox, 2008. <http://www.dropbox.com/>.
- [Eagle and Pentland, 2006a] N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal Ubiquitous Computing*, 10(4):255–268, 2006.
- [Eagle and Pentland, 2006b] Nathan Eagle and Alex Pentland. Eigenbehaviors: Identifying structure in routine. In *Proc. Roy. Soc. A*, 2006.

- [Ekman *et al.*, 2008] Frans Ekman, Ari Keränen, Jouni Karvo, and Jörg Ott. Working day movement model. In *MobilityModels '08: Proceeding of the 1st ACM SIGMOBILE workshop on mobility models*, 2008.
- [Fall, 2003] K. Fall. A delay-tolerant network architecture for challenged internets. In *Intel Research Berkeley, IRB-TR-03-003.*, February 2003.
- [Fall, 2004] K. Fall. Messaging in difficult environments. In *Intel Research Berkeley, IRB-TR-04-019.*, December 2004.
- [Ganchev *et al.*, 2010] Ivan Ganchev, Stanimir Stoyanov, Mairtin O'Droma, Veselina Valkanova, and Damien Meere. Pervasive infostation-based mlearning system. *Systems and Networks Communication, International Conference on*, 0:320–325, 2010.
- [Ghosh *et al.*, 2007] J. Ghosh, S. J. Philip, and C. Qiao. Sociological orbit aware location approximation and routing (solar) in MANET. *Ad Hoc Netw.*, 5:189–209, March 2007.
- [Girardin *et al.*, 2008] Fabien Girardin, Francesco Calabrese, Filippo Dal Fiore, Carlo Ratti, and Josep Blat. Digital footprinting: Uncovering tourists with user-generated content. *Pervasive Computing, IEEE*, 7(4):36–43, Oct.-Dec. 2008.
- [Gnutella, 2000] Gnutella. Gnutella, 2000. <http://www.the-gdf.org/>.
- [Gonzalez *et al.*, 2008] Marta C. Gonzalez, Cesar A. Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008.
- [Goodman *et al.*, 1997] D. J. Goodman, J. Borrs, N. B. Mandayam, and R. D. Yates. Infostations: a new system for data and messaging services. In *Proc. IEEE VTC'97, vol. 2*, pages 969–973, 1997.
- [Grcar *et al.*, 2006] Miha Grcar, Dunja Mladenec, and Marko Grobelnik. User profiling for the web. *Computer Science Information Systems*, 3(2), 2006.
- [Groove, 2007] MS Groove. Microsoft Office Groove 2007, 2007. <http://office.microsoft.com/en-us/groove/>.

- [Hayes and Wilson, 2004] Anna Hayes and David Wilson. Peer-to-peer information sharing in a mobile ad hoc environment. In *Proceedings of the Sixth IEEE Workshop on Mobile Computing Systems and Applications*, 2004.
- [Heinemann *et al.*, 2003] Andreas Heinemann, Jussi Kangasharju, Fernando Lyardet, and Max Mühlhäuser. iClouds – Peer-to-Peer Information Sharing in Mobile Environments. In Harald Kosch, László Böszörményi, and Hermann Hellwagner, editors, *Euro-Par 2003. Parallel Processing, 9th International Euro-Par Conference*, volume 2790 of *Lecture Notes in Computer Science*, pages 1038–1045, Klagenfurt, Austria, 2003. Springer.
- [Henderson *et al.*, 2004] T. Henderson, D. Kotz, and I. Abyzov. The changing usage of a mature campus-wide wireless network. In *MOBICOM*, pages 187–201, 2004.
- [Hsu *et al.*, 2008] W. Hsu, D. Dutta, and A. Helmy. Profile-cast: behavior-aware mobile networking. *SIGMOBILE Mob. Comput. Commun. Rev.*, 12(1):52–54, 2008.
- [Hu and Dittmann, 2009] Liang Hu and Lars Dittmann. Heterogeneous community-based mobility model for human opportunistic network. *IEEE International Conference on Wireless and Mobile Computing, Networking and Communication, Marrakech, Morocco*, 0:465–470, 2009.
- [Hui and Crowcroft, 2007] P. Hui and J. Crowcroft. How small labels create big improvements. In *PERCOMW '07: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 65–70, Washington, DC, USA, 2007. IEEE Computer Society.
- [Hui *et al.*, 2007] P. Hui, E. Yoneki, S.Y. Chan, and J. Crowcroft. Distributed community detection in delay tolerant networks. In *MobiArch '07: Proceedings of first ACM/IEEE international workshop on mobility in the evolving internet architecture*, New York, NY, USA, 2007. ACM.
- [Izquierdo *et al.*, 2009] Luis R. Izquierdo, Segismundo Samuel Izquierdo, Jose Manuel Galan, and Jos Ignacio Santos. Techniques to understand computer simulations: Markov chain analysis. *Journal of Artificial Societies and Social Simulation*, 12:1, 2009.

- [Jacquet *et al.*, 2010] Philippe Jacquet, Bernard Mans, and Georgios Rodolakis. Information propagation speed in mobile and delay tolerant networks. *IEEE Transaction on Information Theory*, 56:5001–5015, October 2010.
- [Jain *et al.*, 2004] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. In *SIGCOMM '04: Proceedings of the 2004 Conference on Applications, technologies, architectures, and protocols for computer communications*, pages 145–158, New York, NY, USA, 2004. ACM.
- [JNotify, 2002] JNotify. JNotify Java API, 2002. <http://jnotify.sourceforge.net/>.
- [Johnson *et al.*, 2007] D. Johnson, Y. Hu, and D. Maltz. The dynamic source routing protocol (dsr). In *RFC 4728.*, February 2007.
- [Jones *et al.*, 2005] Evan P. C. Jones, Lily Li, and Paul A. S. Ward. Practical routing in delay-tolerant networks. In *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 237–243, New York, NY, USA, 2005. ACM.
- [JXTA, 2001] JXTA. *Project JXTA*, 2001. <http://jxta.org/>.
- [Karlin and Taylor, 1981] S. Karlin and H. E. Taylor. *A Second Course in Stochastic Processes*. Academic Press, May 1981.
- [Kernen and Ott, 2007] A. Kernen and J. Ott. Increasing reality for DTN protocol simulations. Technical report, July 2007.
- [Kleinberg, 2007] Jon Kleinberg. Computing: The wireless epidemic. *Nature*, 449(7160):287–288, 2007.
- [Klemm *et al.*, 2003] A. Klemm, C. Lindemann, and O. P. Waldhorst. A special-purpose peer-to-peer file sharing system for mobile ad hoc networks. In *Proc. IEEE Semiannual Vehicular Technology Conference (VTC2003-Fall)*, Orlando, FL, October 2003. IEEE.
- [Klensin, 2001] J. Klensin. RFC 2821: Simple mail transfer protocol, April 2001. <http://www.ietf.org/rfc/rfc2821.txt>.

- [Kortuem *et al.*, 2001] Gerd Kortuem, Jay Schneider, Dustin Preuitt, Thaddeus G. C. Thompson, Stephen Fickas, and Zary Segall. When peer-to-peer comes face-to-face: Collaborative peer-to-peer computing in mobile ad hoc networks. In *P2P '01: Proceedings of the First International Conference on Peer-to-Peer Computing (P2P'01)*, pages 75–89, Washington, DC, USA, 2001. IEEE Computer Society.
- [Kurkowski *et al.*, 2005] S. Kurkowski, T. Camp, and M. Colagrosso. MANET simulation studies: the incredibles. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(4), 2005.
- [Leguay *et al.*, 2005] J. Leguay, T. Friedman, and V. Conan. DTN routing in a mobility pattern space. In *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on delay-tolerant networking*, pages 276–283, New York, NY, USA, 2005. ACM.
- [Leguay *et al.*, 2006] J. Leguay, A. Lindgren, J. Scott, T. Friedman, and J. Crowcroft. Opportunistic content distribution in an urban setting. In *CHANTS '06: Proceedings of the 2006 SIGCOMM workshop on challenged networks*, pages 205–212, New York, NY, USA, 2006. ACM.
- [Liang and Haas, 1999] B. Liang and Z.J. Haas. Predictive distance-based mobility management for PCS networks. *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 3:1377–1384 vol.3, Mar 1999.
- [Librsync, 2005] Librsync. Rsync protocol library, 2005. <http://librsync.sourceforge.net/>.
- [Lindgren *et al.*, 2003] Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, 2003.
- [Malkin, 1998] G. Malkin. RFC 2453: RIP version 2, November 1998. <http://tools.ietf.org/html/rfc2453>.
- [Mcnett and Voelker, 2005] Marvin Mcnett and Geoffrey M. Voelker. Access and mobility of wireless PDA users. *SIGMOBILE Mobile Computing and Communications Review*, 9(2):40–55, April 2005.

- [Mei and Stefa, 2008] Alessandro Mei and Julinda Stefa. SWIM: A simple model to generate small mobile worlds. *CoRR*, abs/0809.2730, 2008.
- [Meijering, 2002] Erik Meijering. A chronology of interpolation: From ancient astronomy to modern signal and image processing. In *Proceedings of the IEEE, VOL. 90, NO. 3*, pages 319–342, March 2002.
- [Miklas *et al.*, 2007] Andrew G. Miklas, Kiran K. Gollu, Kelvin K. W. Chan, Stefan Saroiu, P. Krishna Gummadi, and Eyal de Lara. Exploiting social interactions in mobile systems. In *9th International Conference on Ubiquitous Computing (Ubicomp), Innsbruck, Austria*, pages 409–428, 2007.
- [Mikyong *et al.*, 2006] K. Mikyong, D. Kotz, and S. Kim. Extracting a mobility model from real user traces. In *Proceedings of INFOCOM 2006. 25th IEEE International Conference on Computer Communications*, April 2006.
- [Mining, 2003] Reality Mining. MIT media lab reality mining, 2003. <http://reality.media.mit.edu>.
- [Mockapetris, 1987] P. Mockapetris. RFC 1035: Domain names - implementation and specification, November 1987.
- [Moghadam *et al.*, 2006] A. Moghadam, S. Srinivasan, S. G. Hong, and H. Schulzrinne. Bonjour for Ad-hoc Applications, 2006. <http://bonaha.cvs.sourceforge.net/bonaha>.
- [Moghadam *et al.*, 2008] A. Moghadam, S. Srinivasan, and H. Schulzrinne. 7DS - a modular platform to develop mobile disruption-tolerant applications. In *Next Generation Mobile Applications, Services and Technologies, NGMAST 2008, Wales, UK.*, September 2008.
- [Moy, 1998] J. Moy. RFC 2328: OSPF version 2, April 1998. <http://www.ietf.org/rfc/rfc2328.txt>.
- [Mundur *et al.*, 2006] P. Mundur, S. Lee, and M. Seligman. Routing for data delivery in dynamic networks. *Military Communications Conference. MILCOM 2006, Washington, D.C.*, Oct. 2006.

- [Musolesi and Mascolo, 2006] Mirco Musolesi and Cecilia Mascolo. A community based mobility model for ad hoc network research. In *REALMAN '06: Proceedings of the 2nd international workshop on multi-hop ad hoc networks: from theory to reality*, New York, NY, USA, 2006. ACM.
- [Nioclaïs *et al.*, 2003] Donal Mac Nioclaïs, Denis Chalon, Bruno Richard, and Bruno Richard. Cliques: a transparent, peer-to-peer collaborative file sharing system. In *Proceedings of the 4th international conference on mobile data management (MDM 03)*, pages 21–24, 2003.
- [Norris, 1998] J. R. Norris. *Markov chains*. Cambridge Univ. Press, Cambridge, reprinted edition, 1998.
- [OLPC, 2005] OLPC. One Laptop Per Child, 2005. <http://wiki.laptop.org/go/Home>.
- [ONE, 2007] The ONE. The ONE: The opportunistic network environment simulator, 2007. <http://www.netlab.tkk.fi/tutkimus/dtn/theone>.
- [Osinski *et al.*, 2004] Stanislaw Osinski, Jerzy Stefanowski, and Dawid Weiss. Lingo: Search results clustering algorithm based on singular value decomposition. In *Intelligent Information Systems*, pages 359–368, 2004.
- [Papadopoulou and Schulzrinne, 2000] M. Papadopoulou and H. Schulzrinne. Seven degrees of separation in mobile ad hoc networks. In *IEEE GLOBECOM, San Francisco*, November 2000.
- [Papadopoulou and Schulzrinne, 2001a] Maria Papadopoulou and Henning Schulzrinne. Design and implementation of a peer-to-peer data dissemination and prefetching tool for mobile users. In *First NY Metro Area Networking Workshop*, IBM TJ Watson Research Center, Hawthorne, New York, 2001.
- [Papadopoulou and Schulzrinne, 2001b] Maria Papadopoulou and Henning Schulzrinne. A performance analysis of 7ds a peer-to-peer data dissemination and prefetching tool for mobile users. In *Advances in wired and wireless communications, IEEE Sarnoff Symposium Digest*, Ewing, NJ, 2001.

- [Paul *et al.*, 2008] Sanjoy Paul, Roy Yates, Dipankar Raychaudhuri, and Jim Kurose. The cache-and-forward network architecture for efficient mobile content delivery services in the future internet. In *Proceedings of the First ITU-T Kaleidoscope Academic Conference on Innovations in NGN: Future Network and Services*, pages 319–326, 2008.
- [Perkins *et al.*, 2003] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (aodv) routing. In *RFC 3561.*, July 2003.
- [Raento *et al.*, 2005] M. Raento, A. Oulasvirta, R. Petit, and H. Toivonen. Contextphone: a prototyping platform for context-aware mobile applications. *Pervasive Computing, IEEE*, 4, 2005.
- [Raychaudhuri, 2008] Samik Raychaudhuri. Introduction to monte carlo simulation. In *WSC '08: Proceedings of the 40th Conference on Winter Simulation*, pages 91–100. Winter Simulation Conference, 2008.
- [Rhee *et al.*, 2008] I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong. On the levy-walk nature of human mobility. In *IEEE INFOCOM: 27th IEEE International Conference on Computer Communications. Proceedings*, 2008.
- [RSS,] RSS. The RSS 2.0 specification. <http://blogs.law.harvard.edu/tech/rss>.
- [RSYNC, 2005] RSYNC. Rsync web page, 2005. <http://samba.anu.edu.au/rsync/>.
- [Rubinstein and Kroese,] Reuven Y. Rubinstein and Dirk P. Kroese. *Simulation and the Monte Carlo Method (Wiley Series in Probability and Statistics)*.
- [Schindelhauer, 2006] Christian Schindelhauer. Mobility in wireless networks. In *32nd Annual Conference on Current Trends in Theory and Practice of Informatics, Czech*, 2006.
- [SenseNetworks, 2008] SenseNetworks. Sense networks, 2008. <http://www.sensenetworks.com/>.
- [Shlesinger *et al.*, 1982] Michael F. Shlesinger, Joseph Klafter, and Y. M. Wong. Random walks with infinite spatial and temporal moments. *J. Statist. Phys.*, 27(3):499–512, 1982.

- [Song *et al.*, 2010] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-Lszl Barabasi. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.
- [SQLite, 2005] SQLite. Sqlite database, 2005. <http://www.sqlite.org/>.
- [Sricharan and Vaidehi, 2008] M. Sricharan and V. Vaidehi. A pragmatic analysis of user mobility patterns in macrocellular wireless networks. *Pervasive and Mobile Computing*, 4(5):616–632, October 2008.
- [Srinivasan *et al.*, 2007] S. Srinivasan, A. Moghadam, S. G. Hong, and H. Schulzrinne. 7DS - node cooperation and information exchange in mostly disconnected networks. In *IEEE International Conference on Communication, Glasgow, Scotland*, June 2007.
- [Stafford, 2002] B. Stafford. libesmtplib, 2002. <http://www.stafford.uklinux.net/libesmtplib>.
- [Steinberg and Cheshire, 2010] D. H. Steinberg and S. Cheshire. *Zero Configuration Networking: The Definitive Guide*. O’Reilly Media; 1 edition, March 2010. <http://files.multicastdns.org/draft-cheshire-dnsext-multicastdns.txt>.
- [Su *et al.*, 2007] Jing Su, James Scott, Pan Hui, Jon Crowcroft, Eyal de Lara, Christophe Diot, Ashvin Goel, Menghow Lim, and Eben Upton. Huggle: Seamless networking for mobile applications. In *UbiComp*, volume 4717, pages 391–408. Springer, September 2007.
- [Thakore and Biswas, 2005] D. Thakore and S. Biswas. Routing with persistent link modeling in intermittently connected wireless networks. *Military Communications Conference, 2005. IEEE MILCOM 2005*, pages 461–467 Vol. 1, Oct. 2005.
- [THTTPD, 2005] THTTPD. thttpd, a small-footprint web server, 2005. <http://www.acme.com/software/thttpd/>.
- [Tridgell, 2000] A. Tridgell. Java Rsync implementation, 2000. <http://jarsync.sourceforge.net/>.
- [Vahdat and Becker, 2000] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. In *Technical Report CS-200006, Duke University.*, April 2000.

- [Wang *et al.*, 2007] A. I. Wang, T. Bjornsgard, and K. Saxlund. Peer2Me - rapid application framework for mobile peer-to-peer applications. In *The 2007 International Symposium on Collaborative Technologies and Systems (CTS 2007)*, Orlando, Florida, USA, May 2007.
- [Wang *et al.*, 2009] Pu Wang, Marta C. González, Cesar A. Hidalgo, and Albert-Laszlo Barabasi. Understanding the spreading patterns of mobile phone viruses. *Science*, 324(5930):1071–1076, May 2009.
- [Yoon *et al.*, 2003] J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1312–1321. IEEE, 2003.
- [Zhao *et al.*, 2004] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 187–198, New York, NY, USA, 2004. ACM.
- [Zhao *et al.*, 2005] Wenrui Zhao, Mostafa Ammar, and Ellen Zegura. Multicasting in delay tolerant networks: semantic models and routing algorithms. In *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on delay-tolerant networking*, pages 268–275, New York, NY, USA, 2005. ACM.